6.036/6.862: Introduction to Machine Learning

Lecture: starts Tuesdays 9:35am (Boston time zone)

Course website: introml.odl.mit.edu

Who's talking? Prof. Tamara Broderick

Questions? discourse.odl.mit.edu ("Lecture 11" category)

Materials: Will all be available at course website

Last Time(s)

I. State machines & Markov decision processes (MDPs)
II. Choosing "best" actions
III. Value iteration; Q-learning

Today's Plan

I. Back to supervised learning
II. Sequential data
III. Recurrent neural networks

Some terminology

Some terminology

- *Reinforcement learning* (RL): learning (to maximize rewards) by interacting with the world

Some terminology

- *Reinforcement learning* (RL): learning (to maximize rewards) by interacting with the world
    - Contrast with *supervised learning*

Some terminology

- *Reinforcement learning* (RL): learning (to maximize rewards) by interacting with the world
  - Contrast with *supervised learning*
- *Model-based RL*: uses explicit conception of next state and reward given current state and action

Some terminology

- *Reinforcement learning* (RL): learning (to maximize rewards) by interacting with the world
  - Contrast with *supervised learning*
- *Model-based RL*: uses explicit conception of next state and reward given current state and action
  - “Model” used many different ways in machine learning

Some terminology

- *Reinforcement learning* (RL): learning (to maximize rewards) by interacting with the world
  - Contrast with *supervised learning*
- *Model-based RL*: uses explicit conception of next state and reward given current state and action
  - “Model” used many different ways in machine learning
  - Contrast with *Model-free RL*

Some terminology

- *Reinforcement learning* (RL): learning (to maximize rewards) by interacting with the world
  - Contrast with *supervised learning*
- *Model-based RL*: uses explicit conception of next state and reward given current state and action
  - "Model" used many different ways in machine learning
  - Contrast with *Model-free RL*
- *Q-learning*

Some terminology

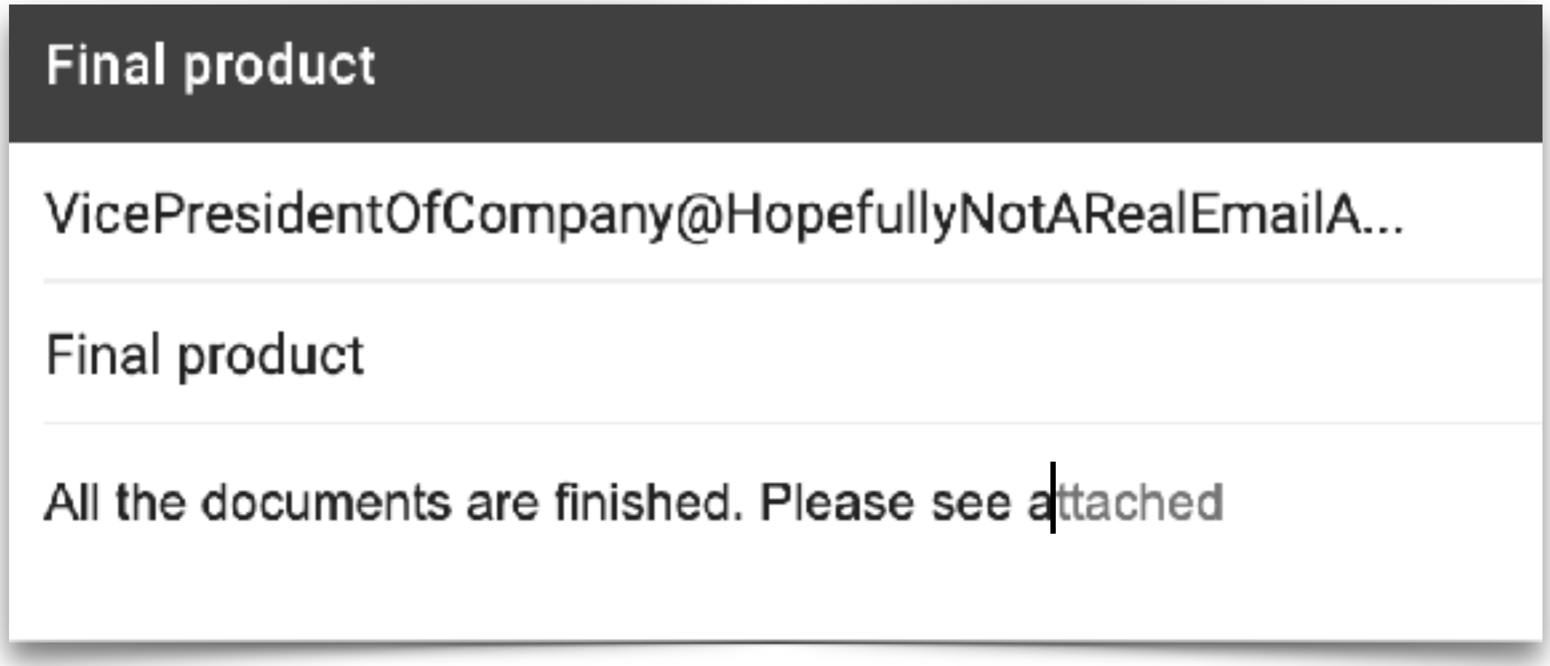
- *Reinforcement learning* (RL): learning (to maximize rewards) by interacting with the world
    - Contrast with *supervised learning*
- *Model-based RL*: uses explicit conception of next state and reward given current state and action
    - "Model" used many different ways in machine learning
    - Contrast with *Model-free RL*
- *Q-learning*
    - Contrast with the Q^* function (expected reward of starting at s, making action a, and then making the "best" action ever after)

Some terminology

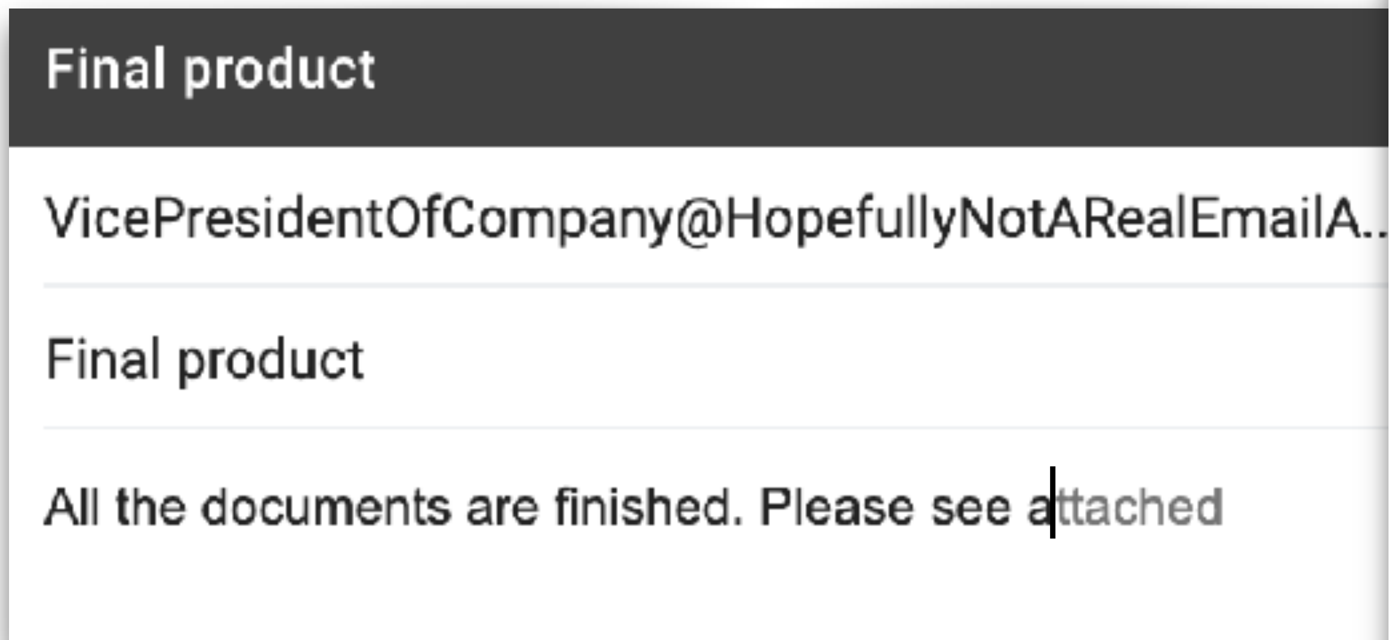
- *Reinforcement learning* (RL): learning (to maximize rewards) by interacting with the world
  - Contrast with *supervised learning*
- *Model-based RL*: uses explicit conception of next state and reward given current state and action
  - "Model" used many different ways in machine learning
  - Contrast with *Model-free RL*
- *Q-learning*
  - Contrast with the *Q** function (expected reward of starting at *s*, making action *a*, and then making the "best" action ever after)
  - Contrast with (any horizon) *value iteration*

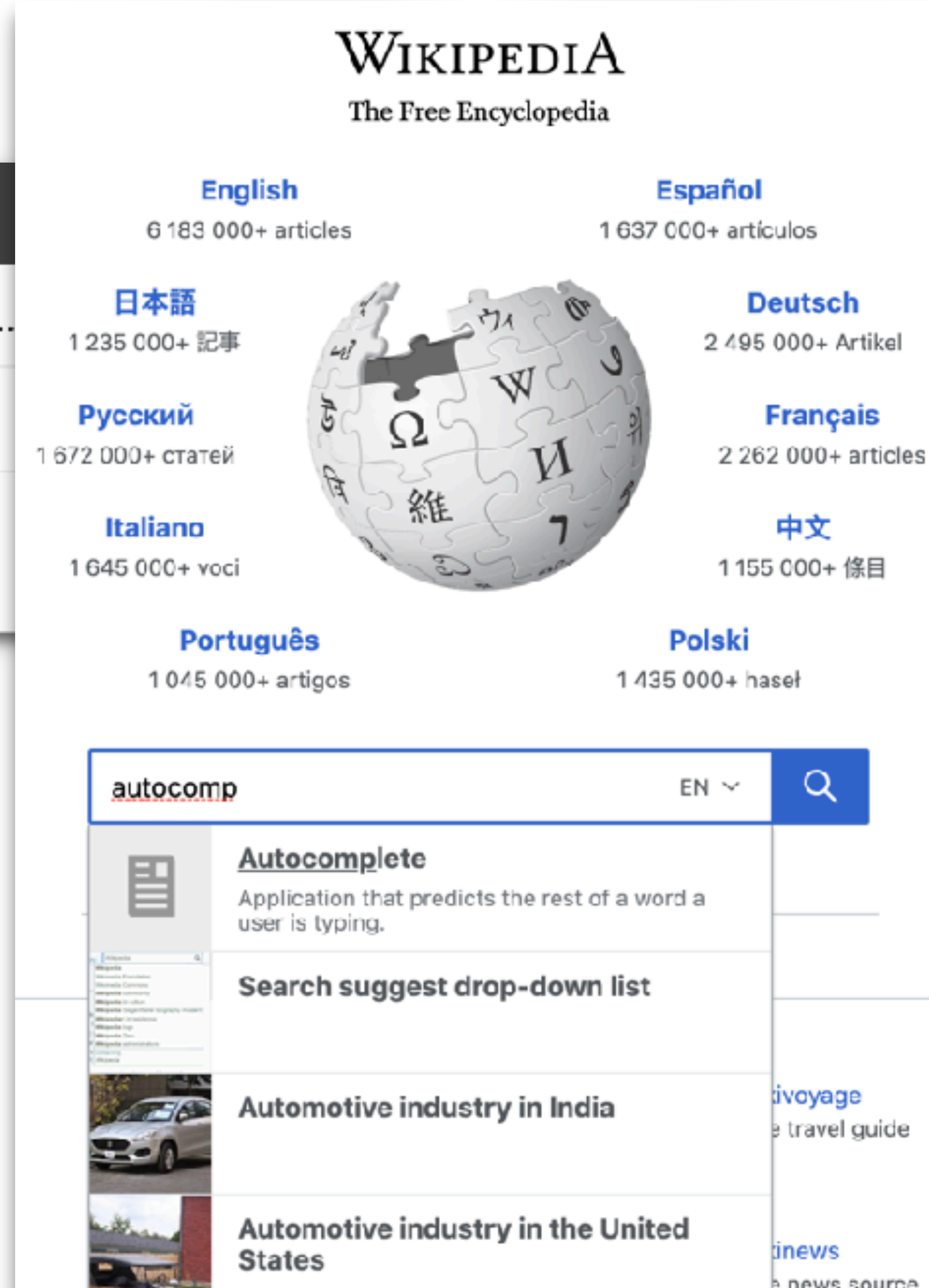
Text prediction

Text prediction

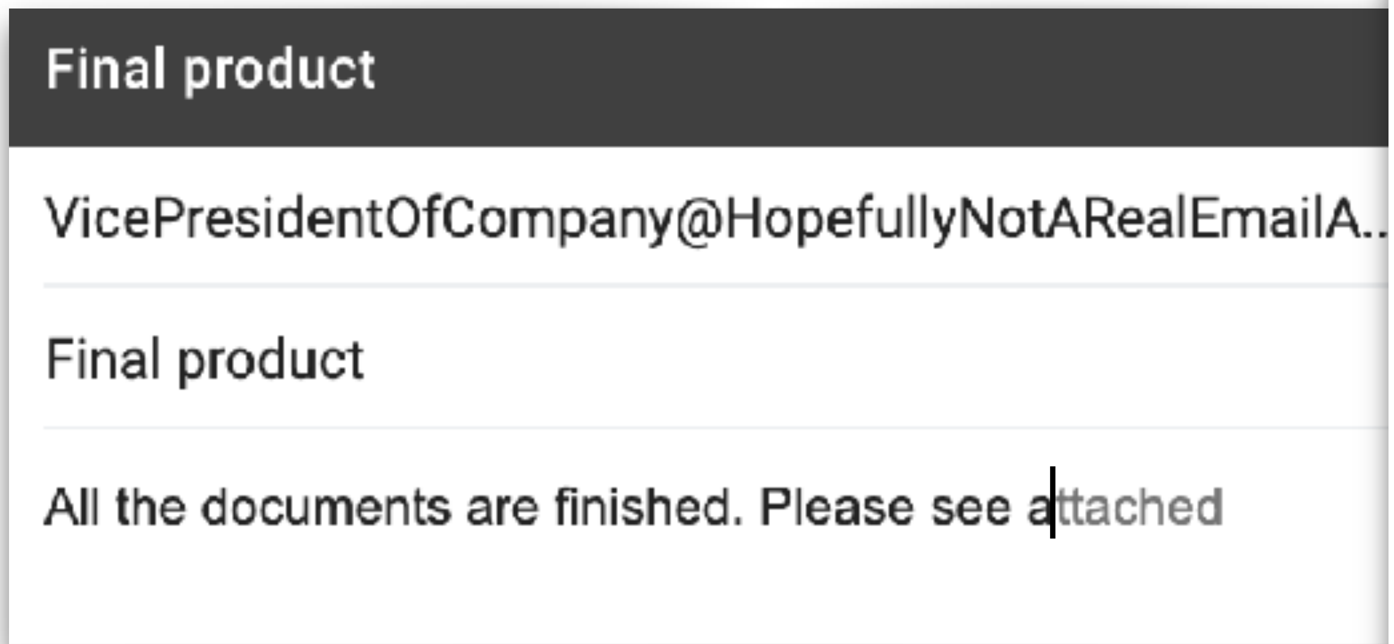


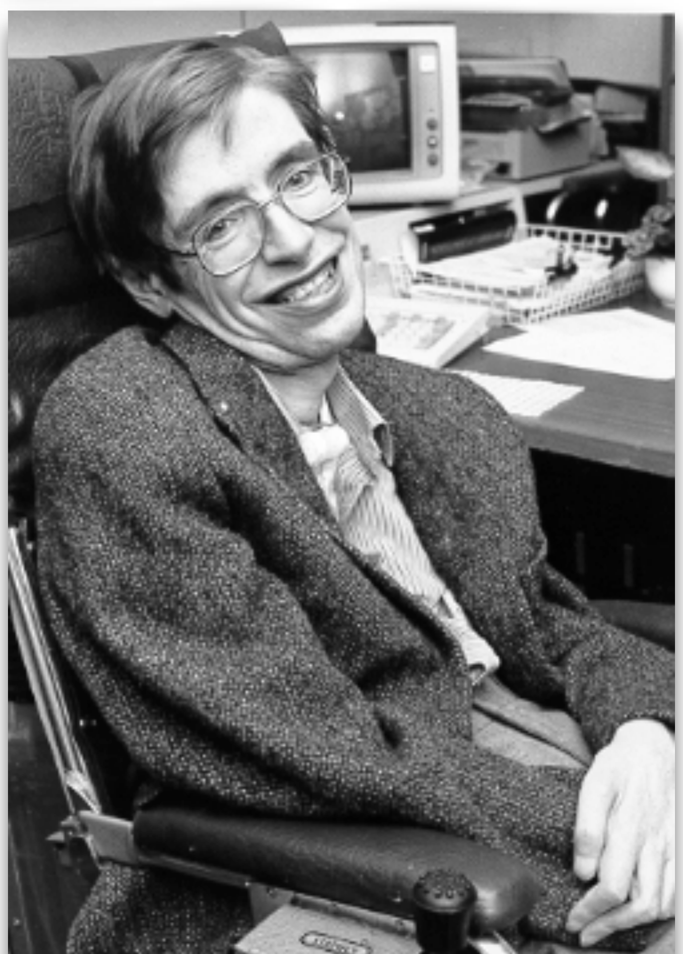
Text prediction

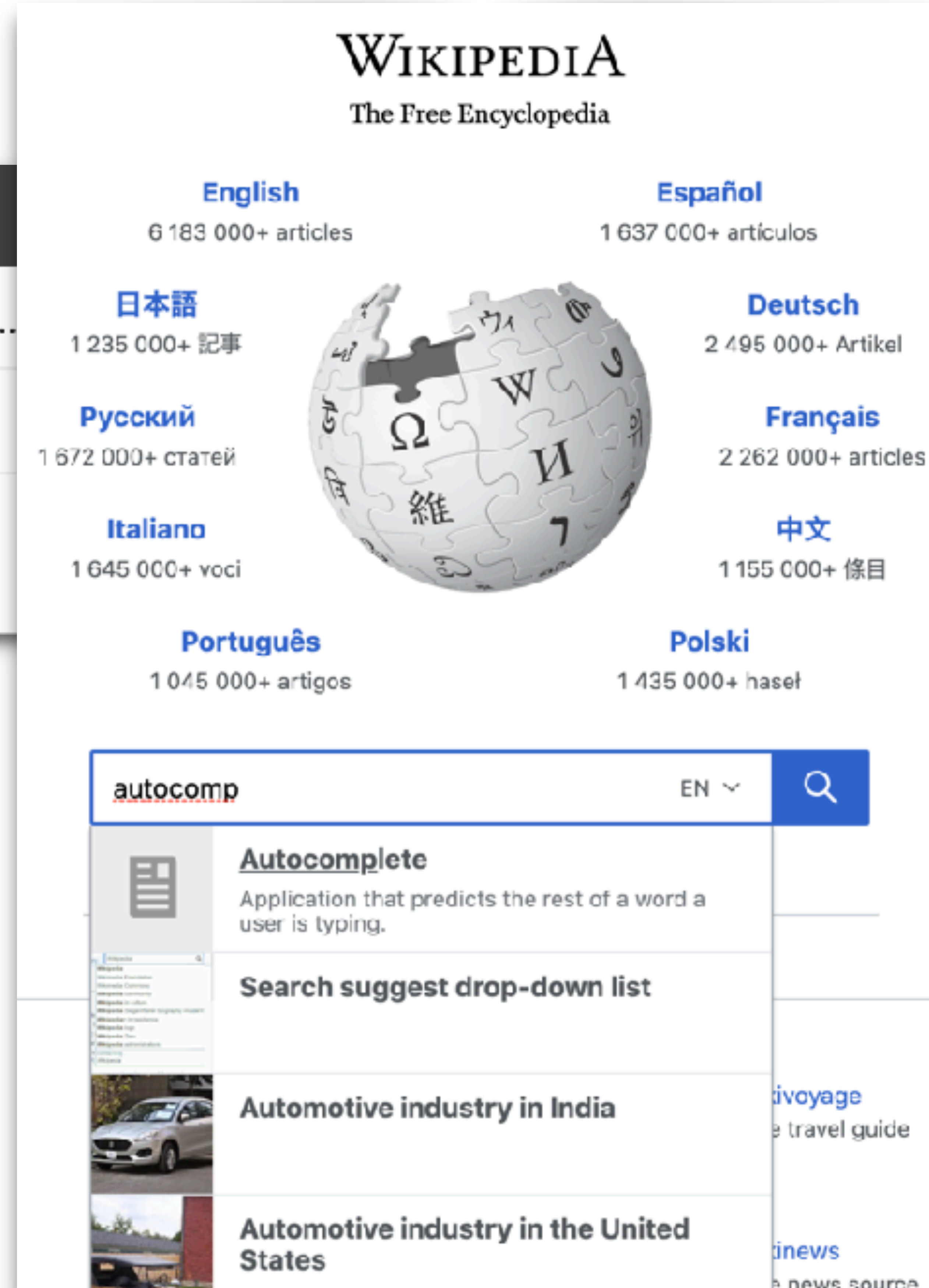




Text prediction

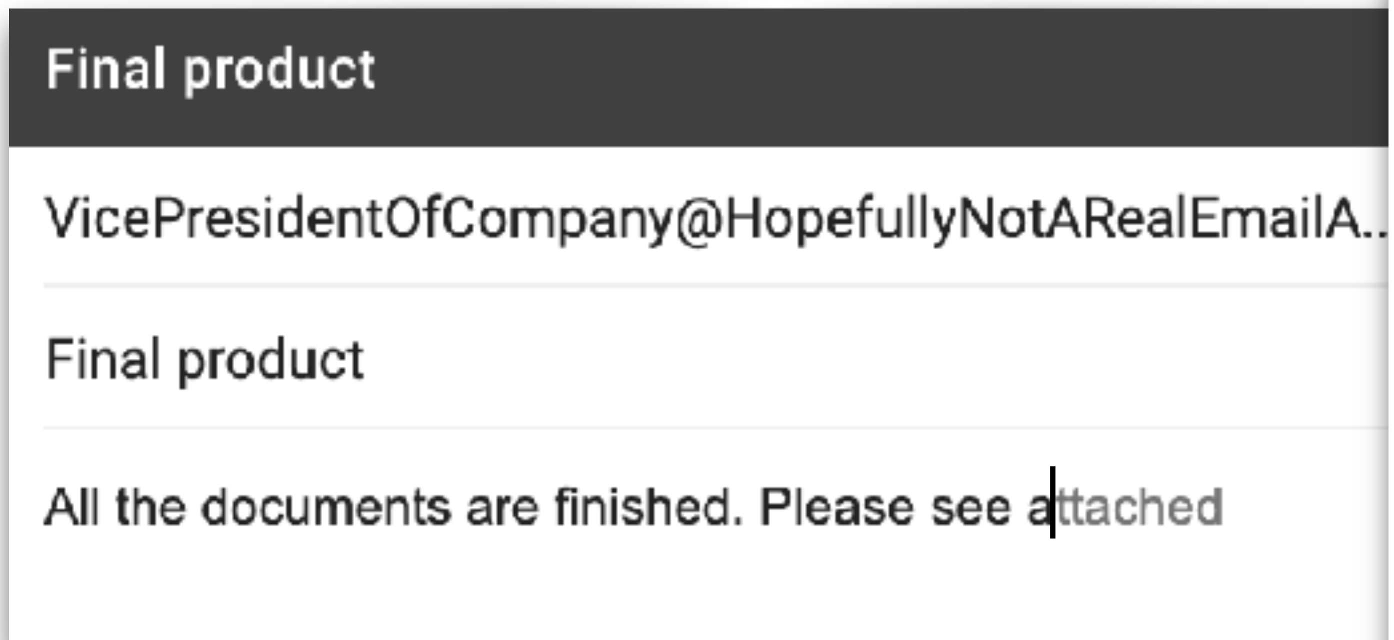


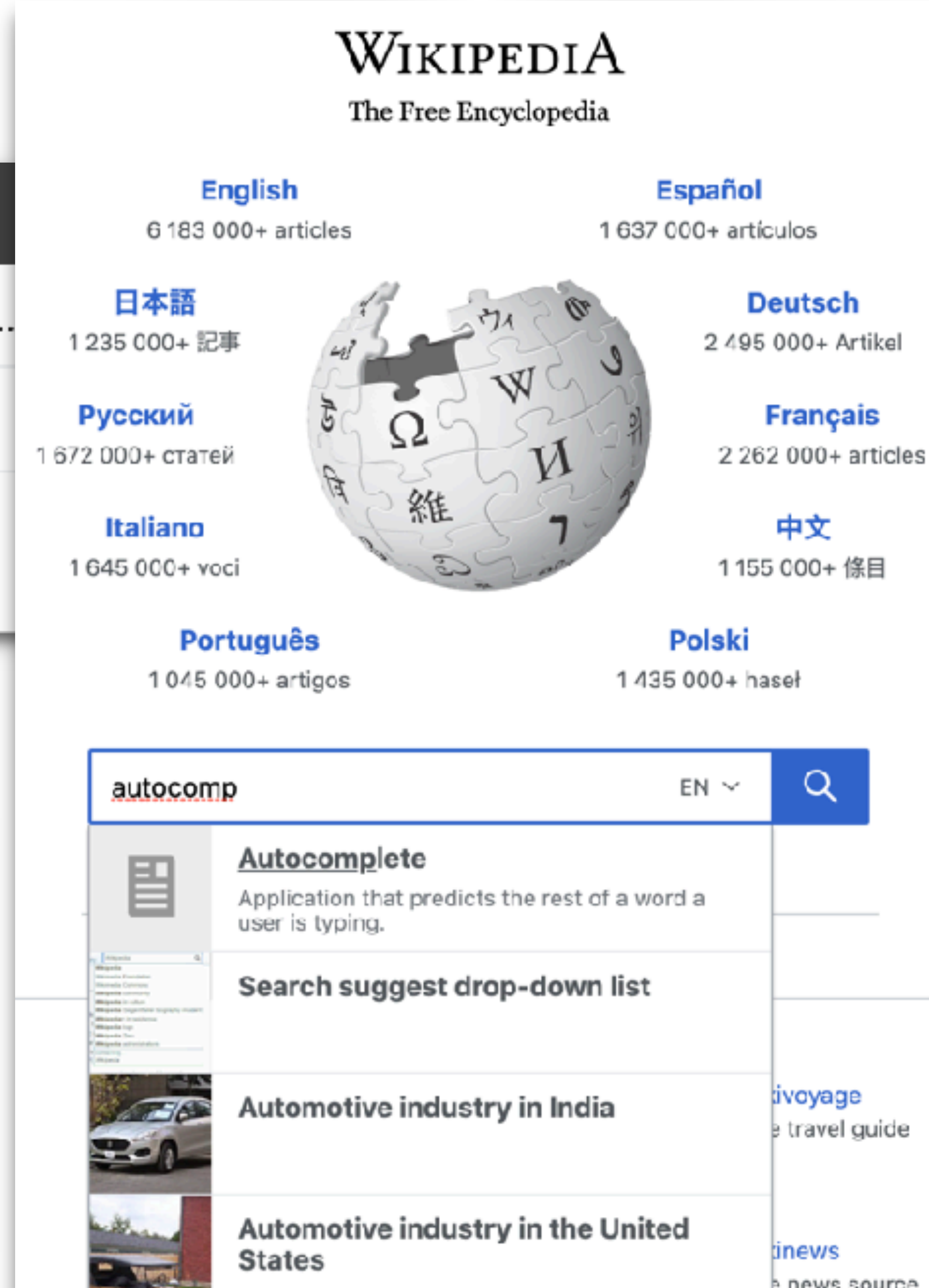




[https://commons.wikimedia.org/wiki/File:Stephen_Hawking.StarChild.jpg]

Text prediction





[https://commons.wikimedia.org/wiki/File:Stephen_Hawking.StarChild.jpg, www.itv.com/news/central/story/2014-08-05/locked-in-syndrome-woman-earns-degree-by-blinking]

Text prediction: supervised learning

Text prediction: supervised learning

- Training data: lots of text

Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

features	label
w	h

Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

features	label
w	h
wh	a

Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

features	label
w	h
wh	a
wha	t

Text prediction: supervised learning

- Training data: lots of text
  - "what happens to a dream deferred"

features	label
w	h
wh	a
wha	t
what	_
what_	h
what_h	a
what_ha	p
what_hap	p
what_happ	e

Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

features	label
w	h
wh	a
wha	t
what	_
what_	h
what_h	a
what_ha	p
what_hap	p
what_happ	e

- Classification with 27 classes

Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

features	label
w	h
wh	a
wha	t
what	_
what_	h
what_h	a
what_ha	p
what_hap	p
what_happ	e

- Classification with 27 classes
- How to featurize?

Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

features	label
w	h
wh	a
wha	t
what	_
what_	h
what_h	a
what_ha	p
what_hap	p
what_happ	e

- Classification with 27 classes
- How to featurize?
- Idea: use all previous characters.

Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

features	label
w	h
wh	a
wha	t
what	_
what_	h
what_h	a
what_ha	p
what_hap	p
what_happ	e

- Classification with 27 classes
- How to featurize?
- Idea: use all previous characters. But so far we’ve said $x^{(i)} \in \mathbb{R}^d$; i.e. fixed dimension

Text prediction: supervised learning

- Training data: lots of text
  - "what happens to a dream deferred"

features	label
w	h
wh	a
wha	t
what	_
what_	h
what_h	a
what_ha	p
what_hap	p
what_happ	e

- Classification with 27 classes
- How to featurize?
- Idea: use all previous characters. But so far we've said $x^{(i)} \in \mathbb{R}^d$; i.e. fixed dimension

Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

features	label
w	h
wh	a
wha	t
what	_
what_	h
what_h	a
what_ha	p
what_hap	p
what_happ	e

- Classification with 27 classes
- How to featurize?
- Idea: use all previous characters. But so far we’ve said $x^{(i)} \in \mathbb{R}^d$; i.e. fixed dimension

Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

features	label
w	h
wh	a
wha	t
what	_
what_	h
what_h	a
what_ha	p
what_hap	p
what_happ	e

- Classification with 27 classes
- How to featurize?
- Idea: use all previous characters. But so far we’ve said $x^{(i)} \in \mathbb{R}^d$; i.e. fixed dimension
- Idea: just use last character. But lose info

Text prediction: supervised learning

- Training data: lots of text
  - "what happens to a dream deferred"

features	label
w	h
wh	a
wha	t
what	_
what_	h
what_h	a
what_ha	p
what_hap	p
what_happ	e

- Classification with 27 classes
- How to featurize?
- Idea: use all previous characters. But so far we've said $x^{(i)} \in \mathbb{R}^d$; i.e. fixed dimension
- Idea: just use last character. But lose info
- Idea: use last *m* characters

Can express as a state machine

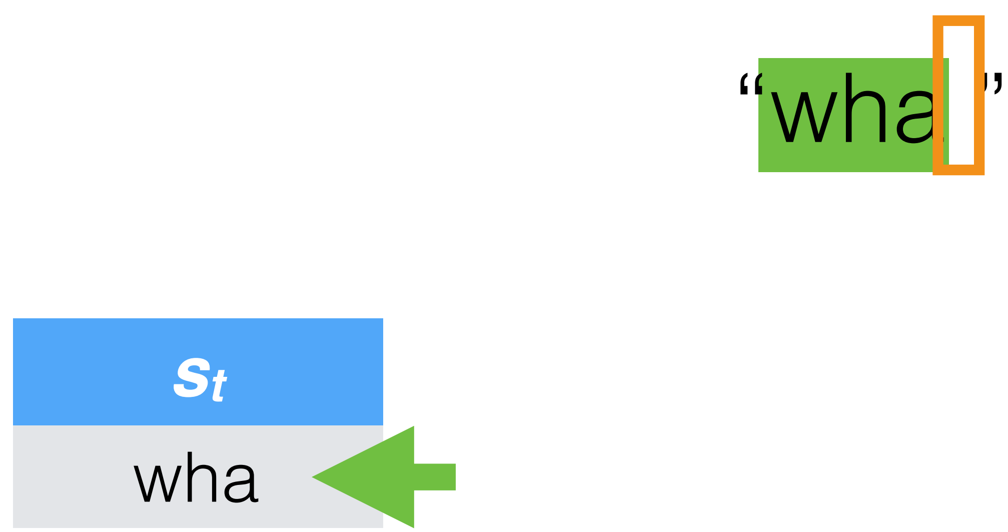
“wha”

Can express as a state machine

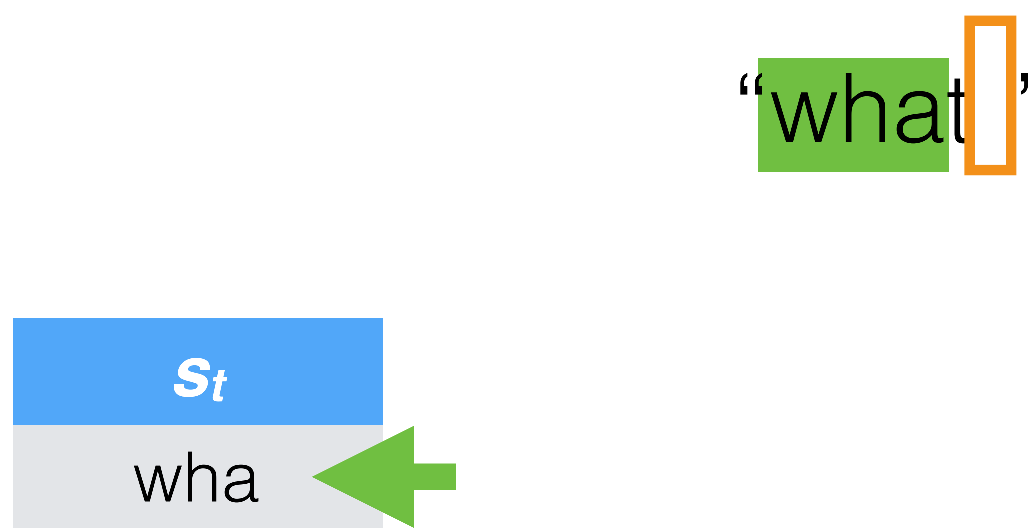
“wha”

Can express as a state machine

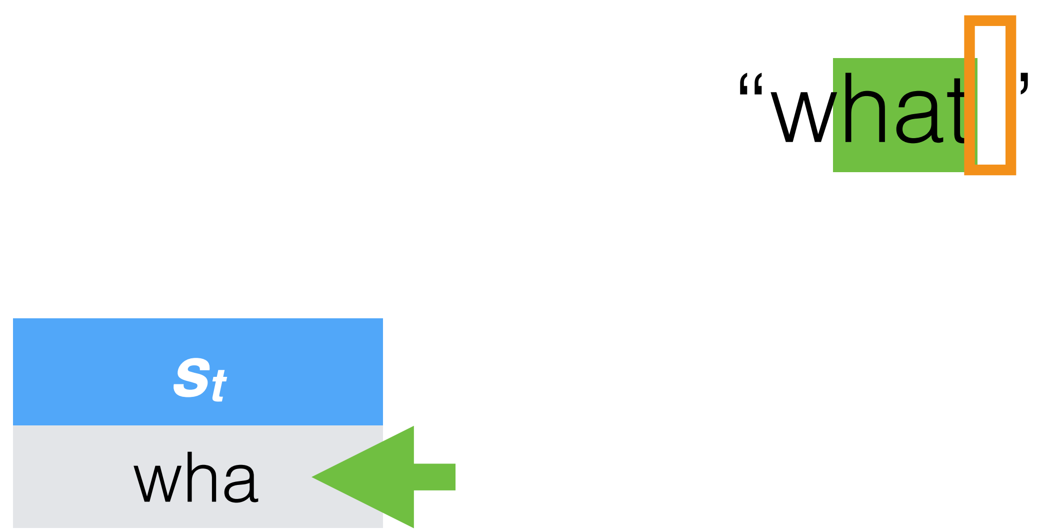
Can express as a state machine



Can express as a state machine



Can express as a state machine



Can express as a state machine

“what ”

x_t	s_t
	wha

t

Can express as a state machine

“what ”

x_t	s_t
	wha
t	hat

Can express as a state machine

"what_ "

x_t	s_t
	wha
t	hat

Can express as a state machine

x_t	s_t
	wha
t	hat

Can express as a state machine

“what_ ”

x_t	s_t
	wha
t	hat
_	at_

Can express as a state machine

“what happens to a dream deferred”

x_t	s_t
	wha
t	hat
_	at_
h	t_h

Can express as a state machine

- Recall state machines:

"what happens to a dream deferred"

x_t	s_t
	wha
t	hat
_	at_
h	t_h

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$

“what happens to a dream deferred”

x_t	s_t
	wha
t	hat
_	at_
h	t_h

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
- Example:
  - All ordered *m* characters

“what happens to a dream deferred”

x_t	s_t
	wha
t	hat
_	at_
h	t_h

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
- Example:
  - All ordered *m* characters

"what happens to a dream deferred"

x_t	s_t
	wha
t	hat
_	at_
h	t_h

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
- Example:
  - All ordered *m* characters
  - All characters

"what happens to a dream deferred"

x_t	s_t
	wha
t	hat
_	at_
h	t_h

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
- Example:
  - All ordered *m* characters
  - All characters

“what happens to a dream deferred”

x_t	s_t
	wha
t	hat
_	at_
h	t_h

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
- Example:
  - All ordered m characters
  - All characters

"^what happens to a dream deferred"

x_t	s_t
	wha
t	hat
_	at_
h	t_h

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
- Example:
  - All ordered *m* characters
  - All characters

"^what happens to a dream deferred"

x_t	s_t
	wha
t	hat
_	at_
h	t_h

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state

- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters

"^what happens to a dream deferred"

x_t	s_t
	wha
t	hat
_	at_
h	t_h

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters

x_t	s_t
	^^^
^	^^^
w	^^w
h	^wh
a	wha
t	hat
_	at_

"^what happens to a dream deferred"

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters

x_t	s_t
	^^^
^	^^^
w	^^w
h	^wh
a	wha
t	hat
_	at_

"^what happens to a dream deferred"

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters

x_t	s_t
	^^^
^	^^^
w	^^w
h	^wh
a	wha
t	hat
_	at_

"^what happens to a dream deferred"

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters

x_t	s_t
	^^^
^	^^^
w	^^w
h	^wh
a	wha
t	hat
_	at_

"^what happens to a dream deferred"

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters

x_t	s_t
	^^^
^	^^^
w	^^w
h	^wh
a	wha
t	hat
_	at_

"^what happens to a dream deferred"

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters

x_t	s_t
	^^^
^	^^^
w	^^w
h	^wh
a	wha
t	hat
_	at_

"^what happens to a dream deferred"

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters

t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

"^what happens to a dream deferred"

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters

t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

"^what happens to a dream deferred"

Can express as a state machine

- Recall state machines:
    - Set of possible states $\mathcal{S}$
    - Set of possible inputs $\mathcal{X}$
    - Initial state
    - Transition function $f(s,x)$
- Example:
    - All ordered *m* characters
    - All characters
    - *m* start characters
    - Update last *m* chars

t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

"^what happens to a dream deferred"

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters
  - Update last *m* chars

t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

"^what happens to a dream deferred"

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters
  - Update last *m* chars
  - All vectors of char probs

t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

"^what happens to a dream deferred"

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $g(s)$

t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters
  - Update last *m* chars
  - All vectors of char probs

"^what happens to a dream deferred"

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $g(s)$

t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters
  - Update last *m* chars
  - All vectors of char probs
  - Multi-class linear classifier

"^what happens to a dream deferred"

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $g(s)$

t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters
  - Update last *m* chars
  - All vectors of char probs
  - Multi-class linear classifier
- $x^{(1)}$: "^what happens to a dream deferred"

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $g(s)$

t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters
  - Update last *m* chars
  - All vectors of char probs
  - Multi-class linear classifier

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

Can express as a state machine

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function *f*(*s*,*x*)
  - Set of possible outputs
  - Output function *g*(*s*)

t	$x^{(1)}_t$	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- Example:
  - All ordered *m* characters
  - All characters
  - *m* start characters
  - Update last *m* chars
  - All vectors of char probs
  - Multi-class linear classifier

- $x^{(1)}$: “^what happens to a dream deferred”
- $x^{(2)}$: “^if you can keep your head when all about you”
- $x^{(3)}$: “^you may write me down in history”

Can express as a state machine

t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

Can express as a state machine

s_0

t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

Can express as a state machine

s_0

x_1

t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

Can express as a state machine

$s_0 \rightarrow$

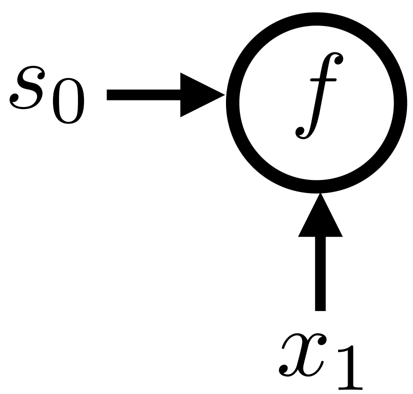
$\uparrow$

x_1

t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

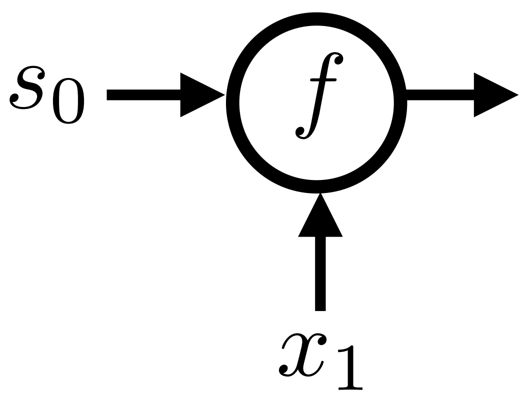
Can express as a state machine



t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

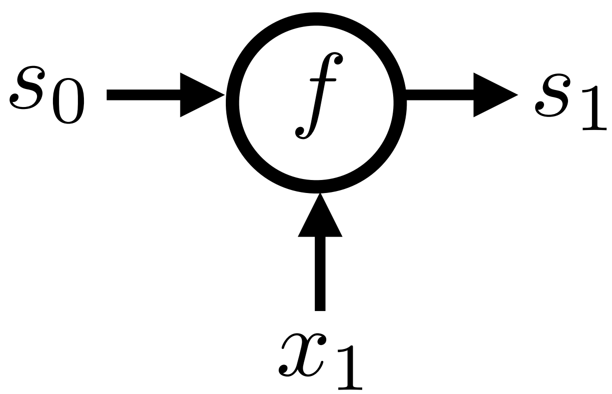
Can express as a state machine



t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

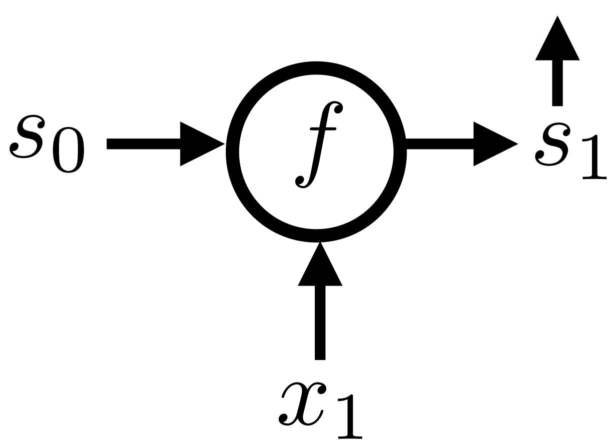
Can express as a state machine



t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

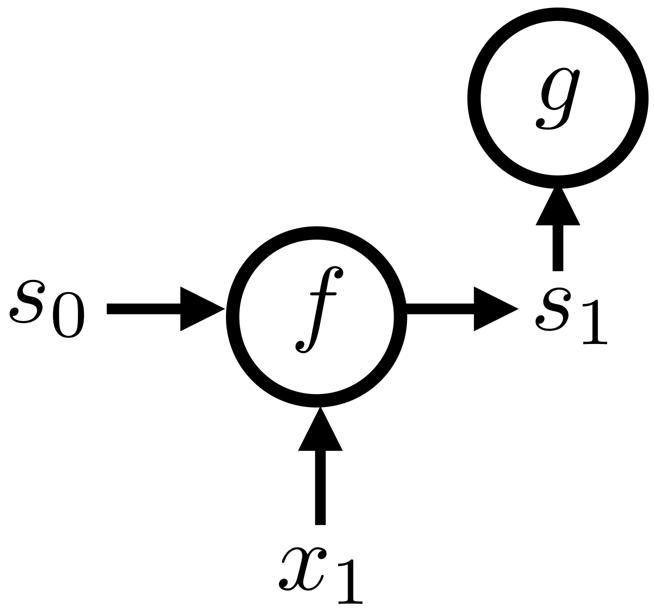
Can express as a state machine



t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

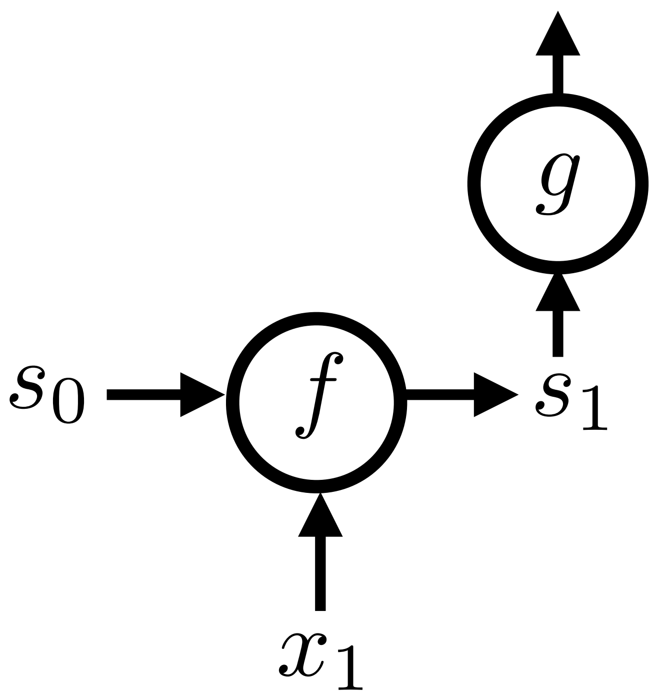
Can express as a state machine



t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

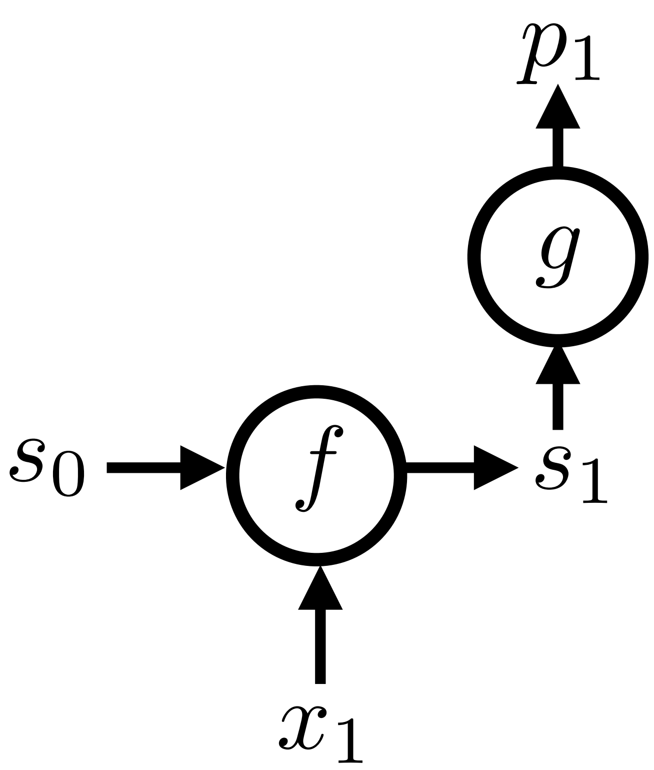
Can express as a state machine



t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

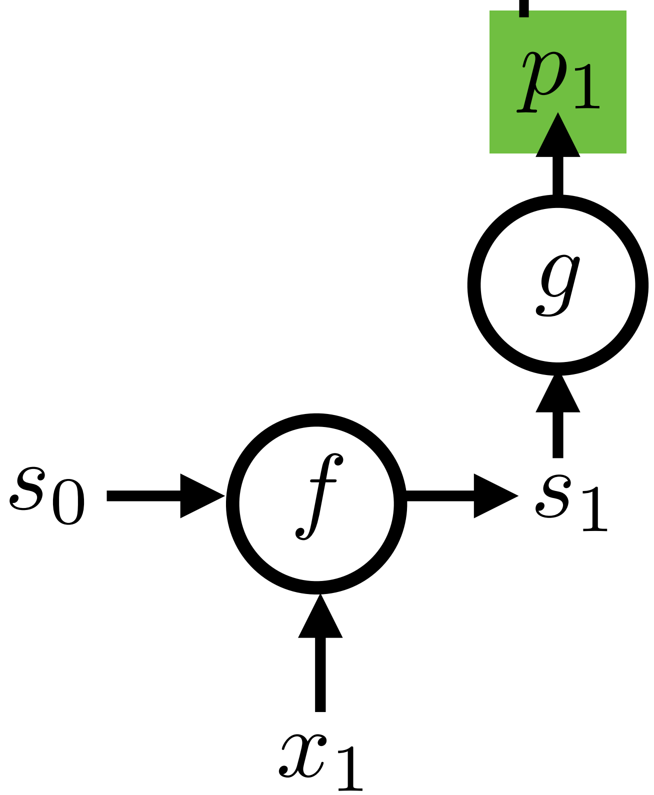
Can express as a state machine



t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

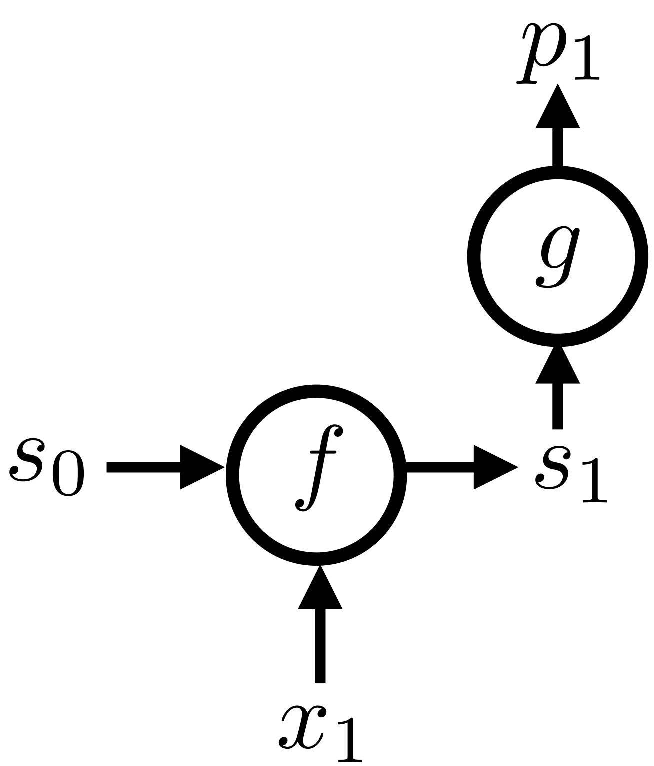
Can express as a state machine



t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: “^what happens to a dream deferred”
- $x^{(2)}$: “^if you can keep your head when all about you”
- $x^{(3)}$: “^you may write me down in history”

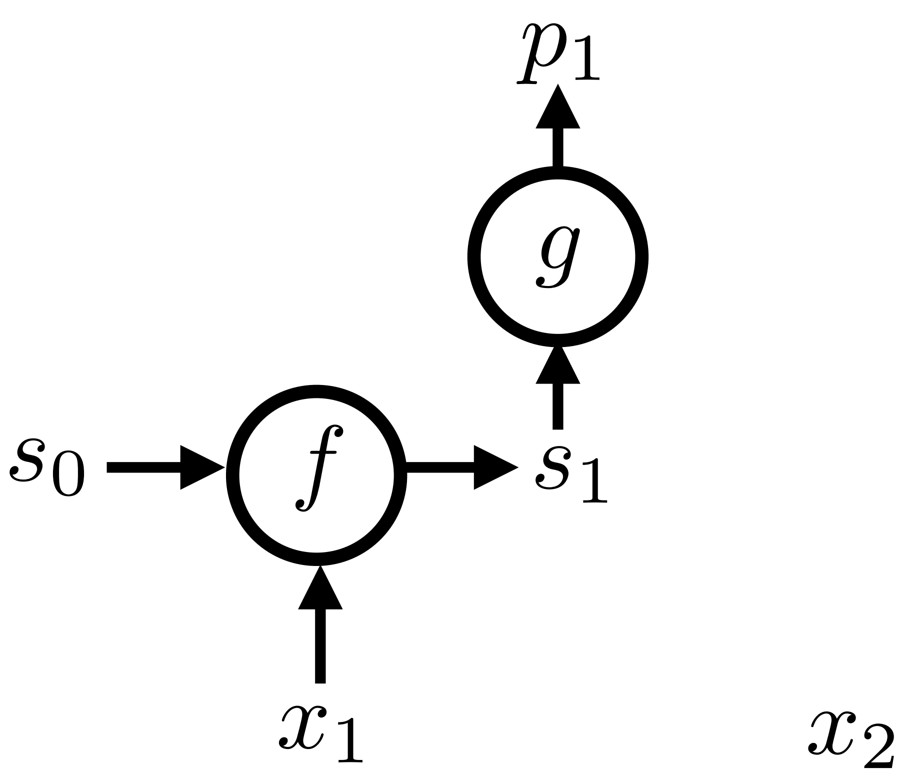
Can express as a state machine



t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

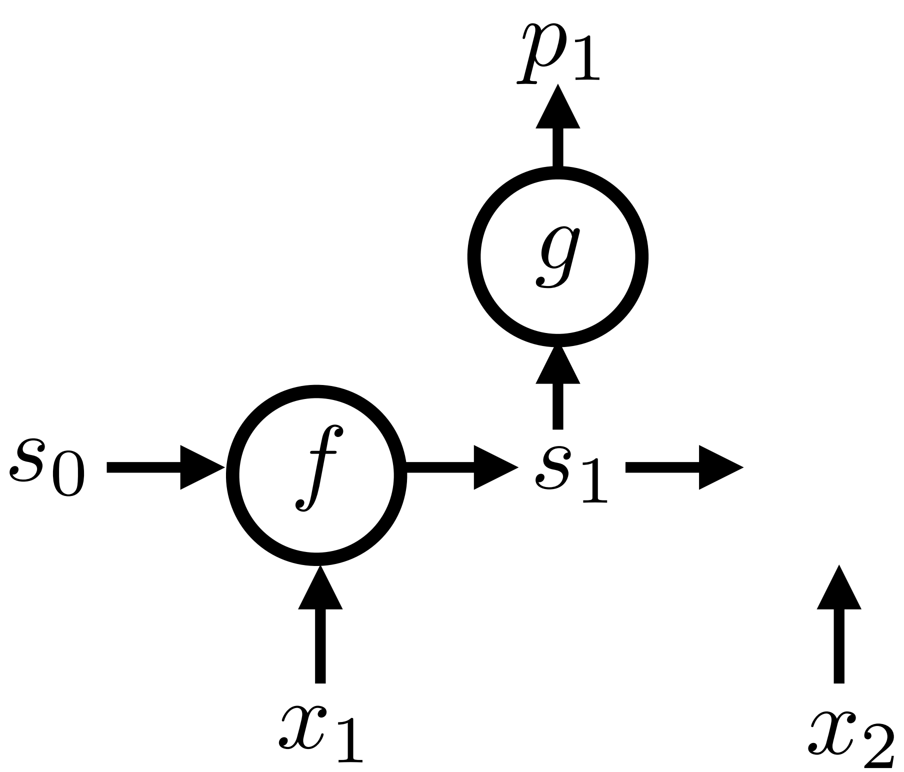
Can express as a state machine



t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

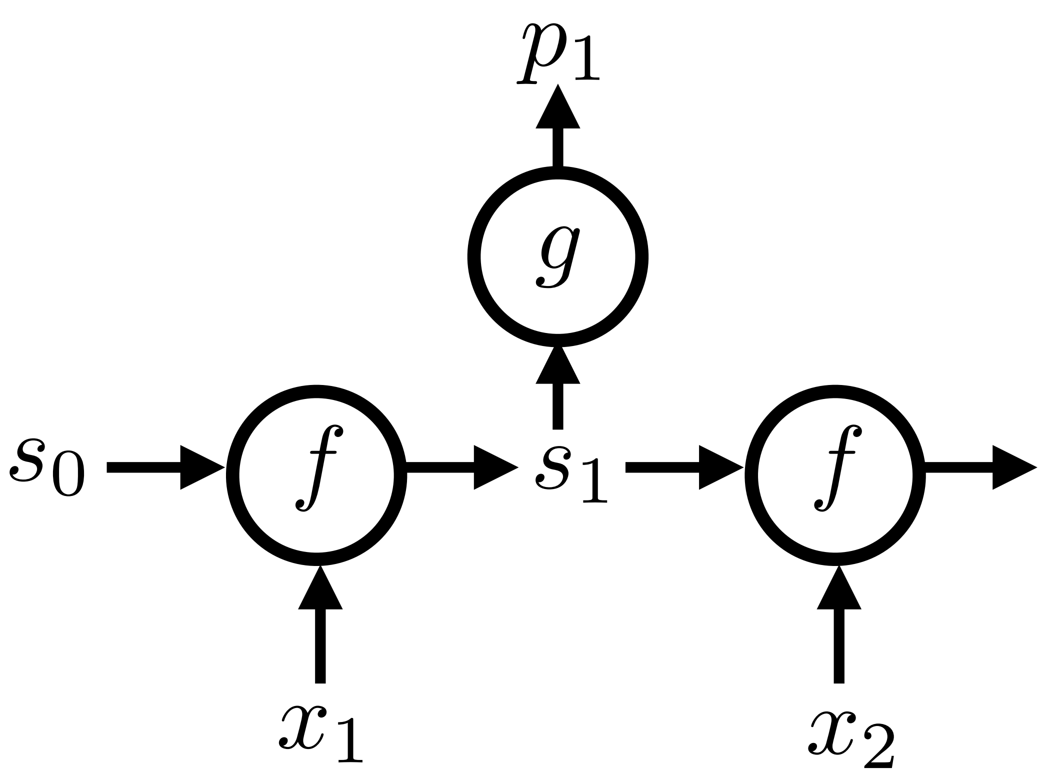
Can express as a state machine



t	*x_t*	*s_t*
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

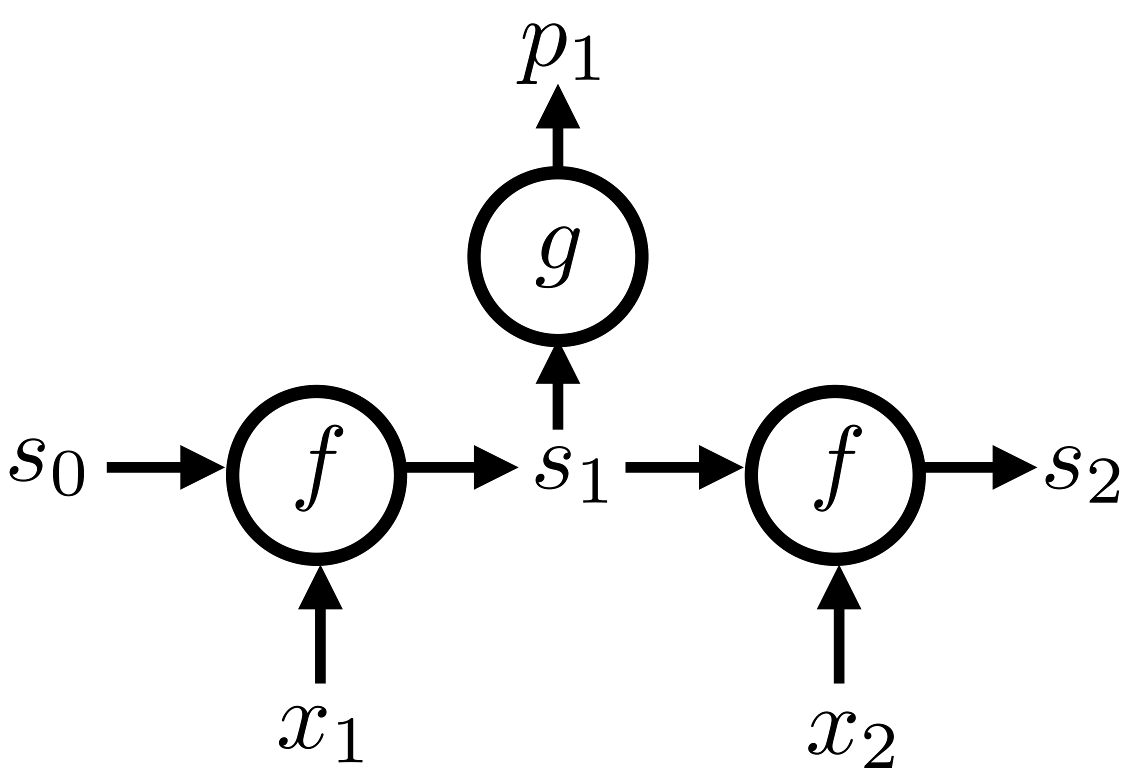
Can express as a state machine



t	*x_t*	*s_t*
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

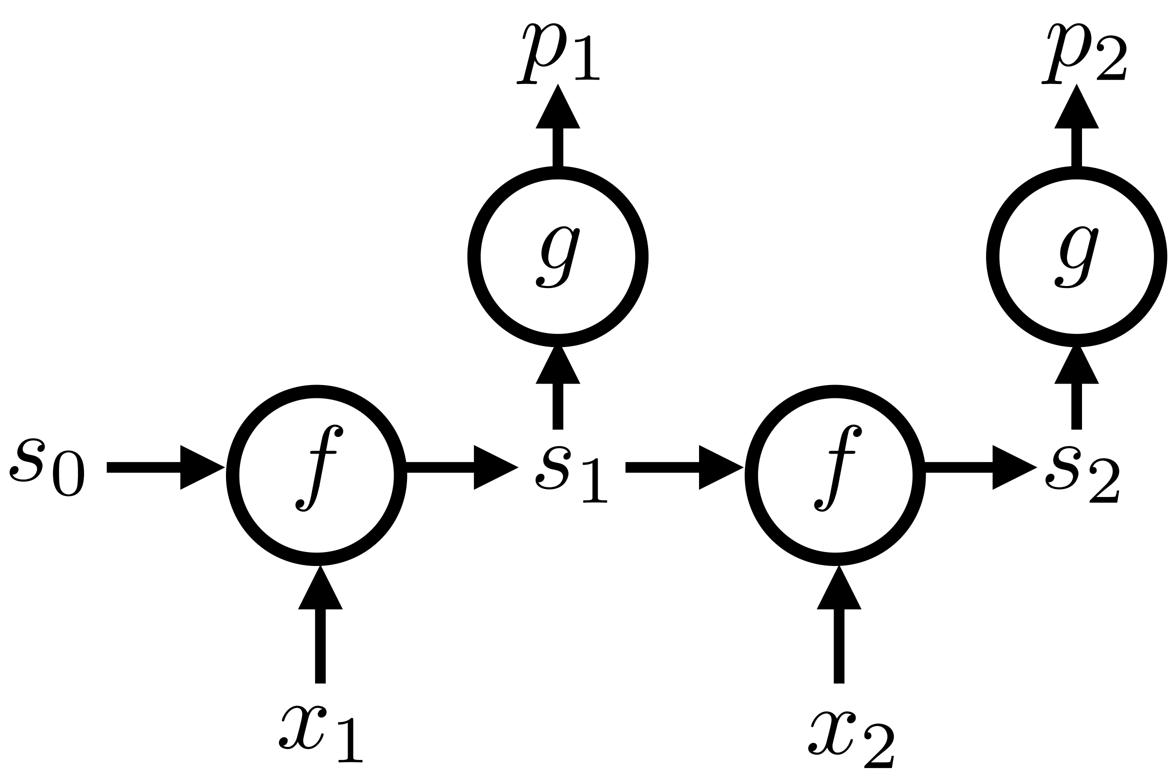
Can express as a state machine



t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

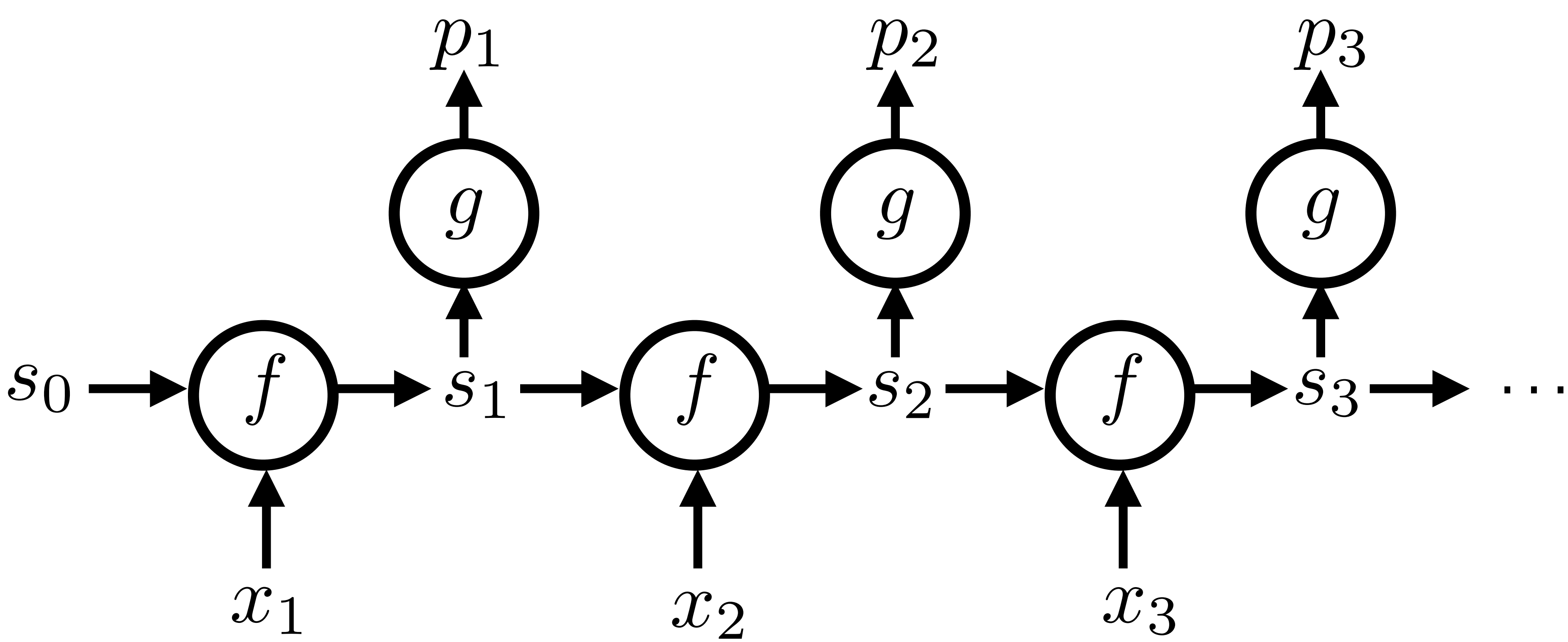
Can express as a state machine



t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

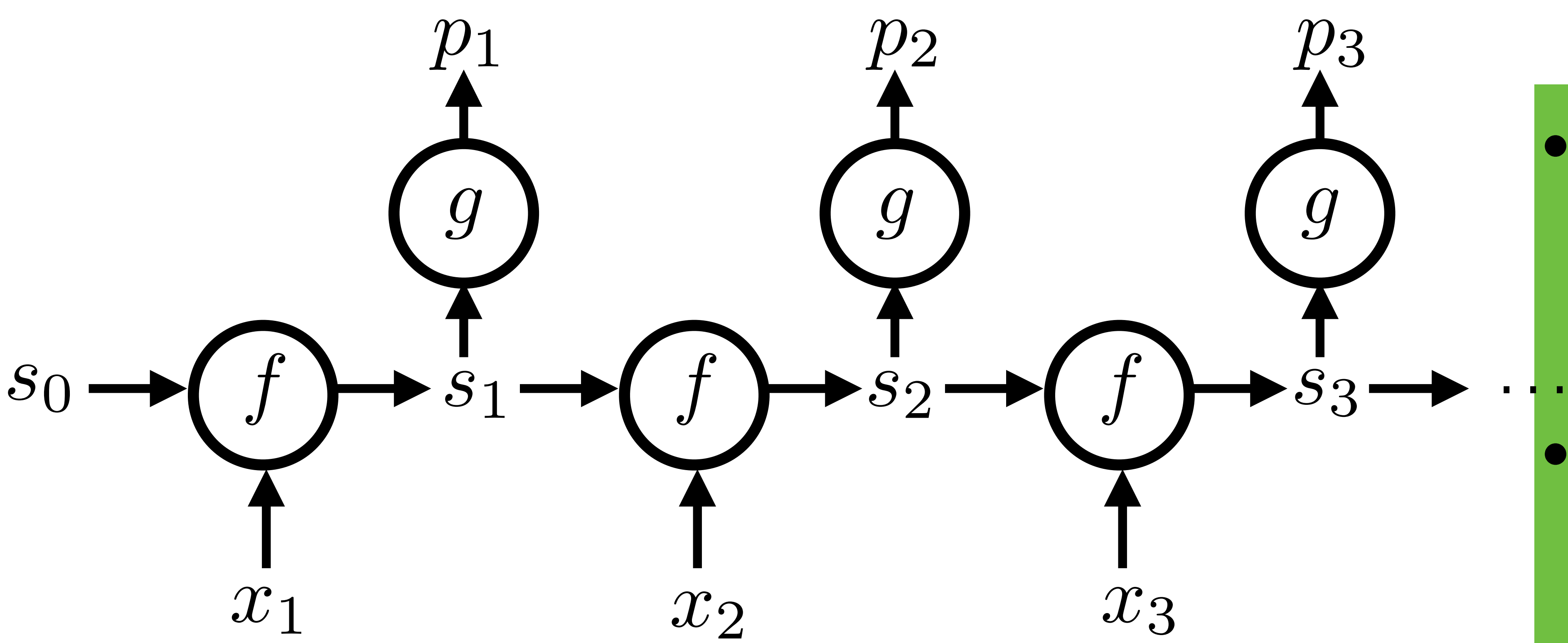
Can express as a state machine



t	***x**$_t$*	***s**$_t$*
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: “^what happens to a dream deferred”
- $x^{(2)}$: “^if you can keep your head when all about you”
- $x^{(3)}$: “^you may write me down in history”

Can express as a state machine

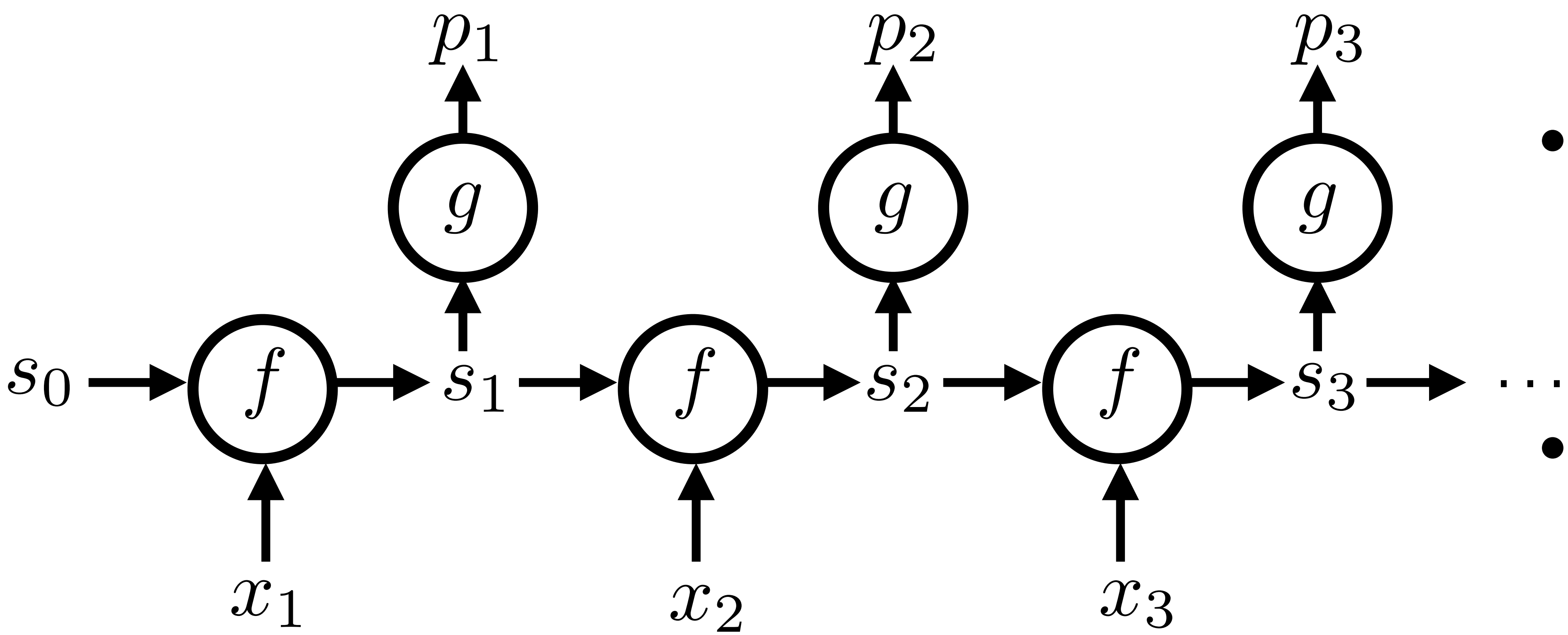


- *m*: number of characters in the context
- *v*: number of characters in the alphabet

t	x_t	s_t
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

Can express as a state machine

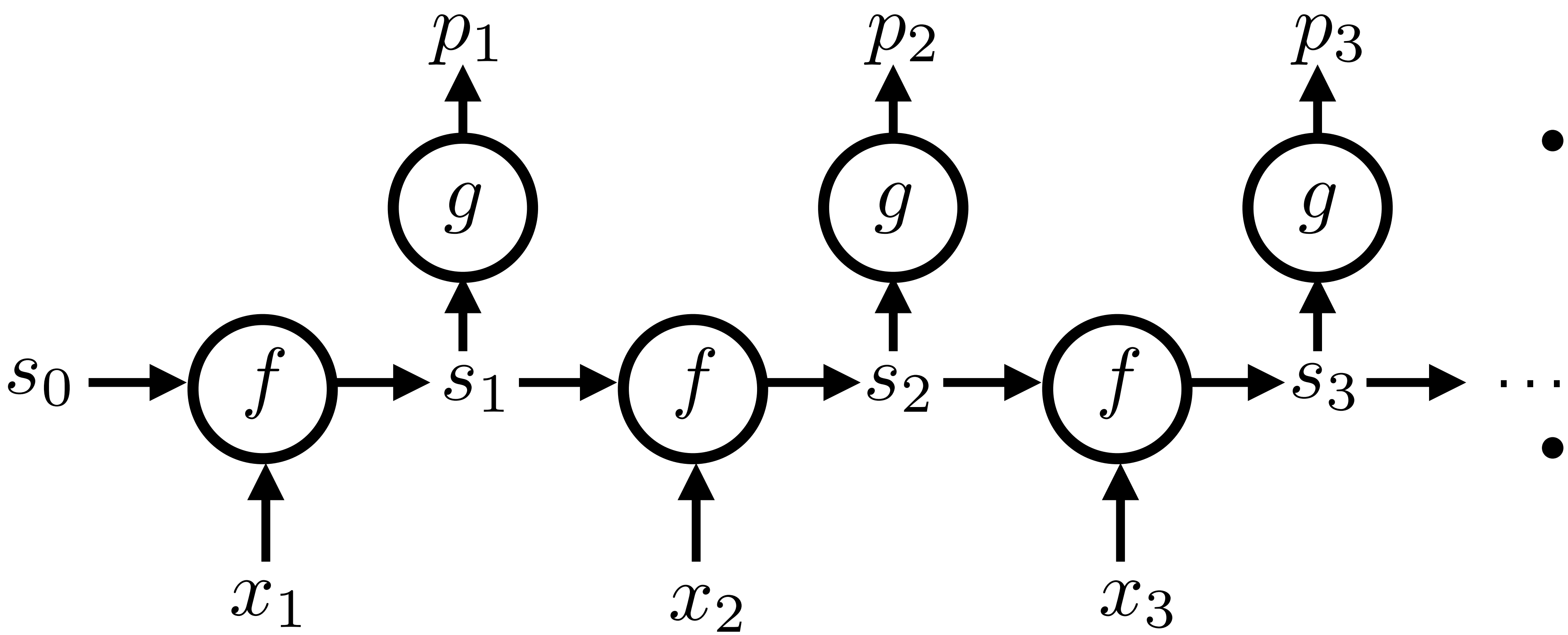


- *m*: number of characters in the context
- *v*: number of characters in the alphabet

t	*x*$_t$	*s*$_t$
0		^^^
1	^	^^^
2	w	^^w
3	h	^wh
4	a	wha
5	t	hat
6	_	at_

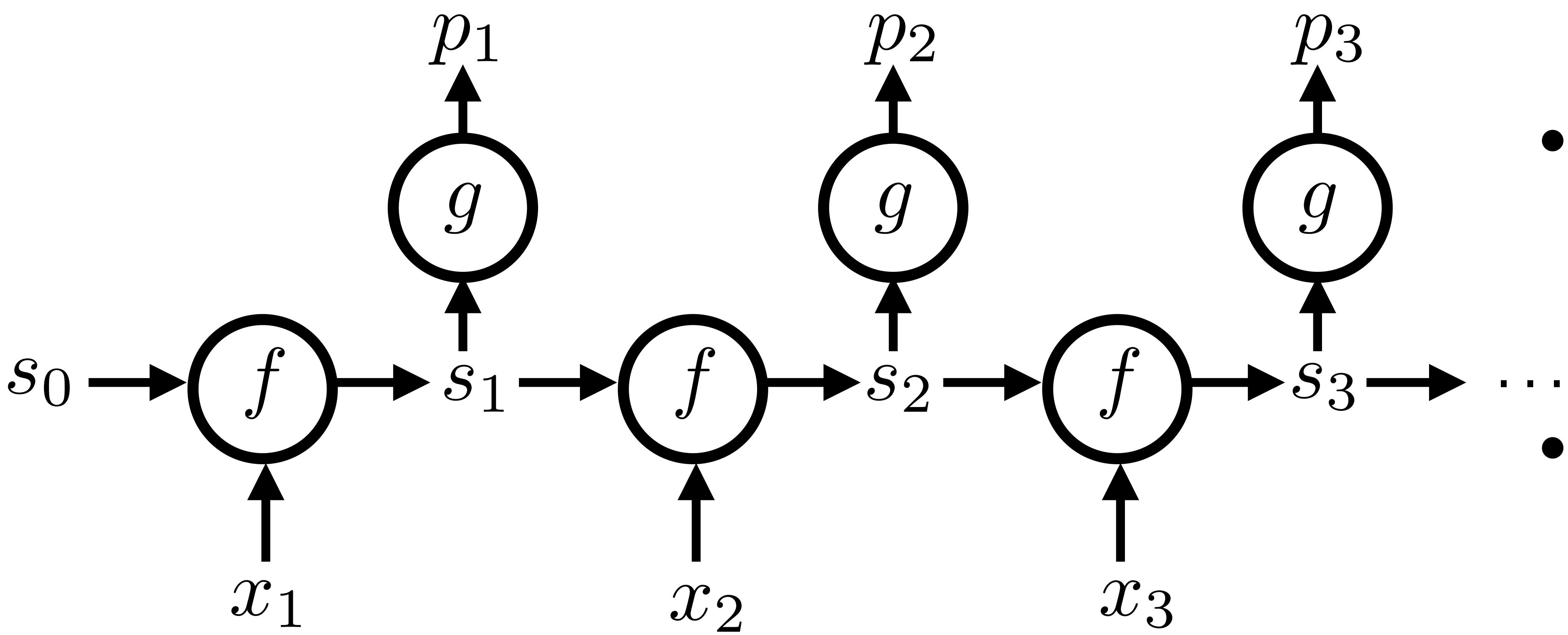
- $x^{(1)}$: "^what happens to a dream deferred"
- $x^{(2)}$: "^if you can keep your head when all about you"
- $x^{(3)}$: "^you may write me down in history"

Can express as a state machine



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

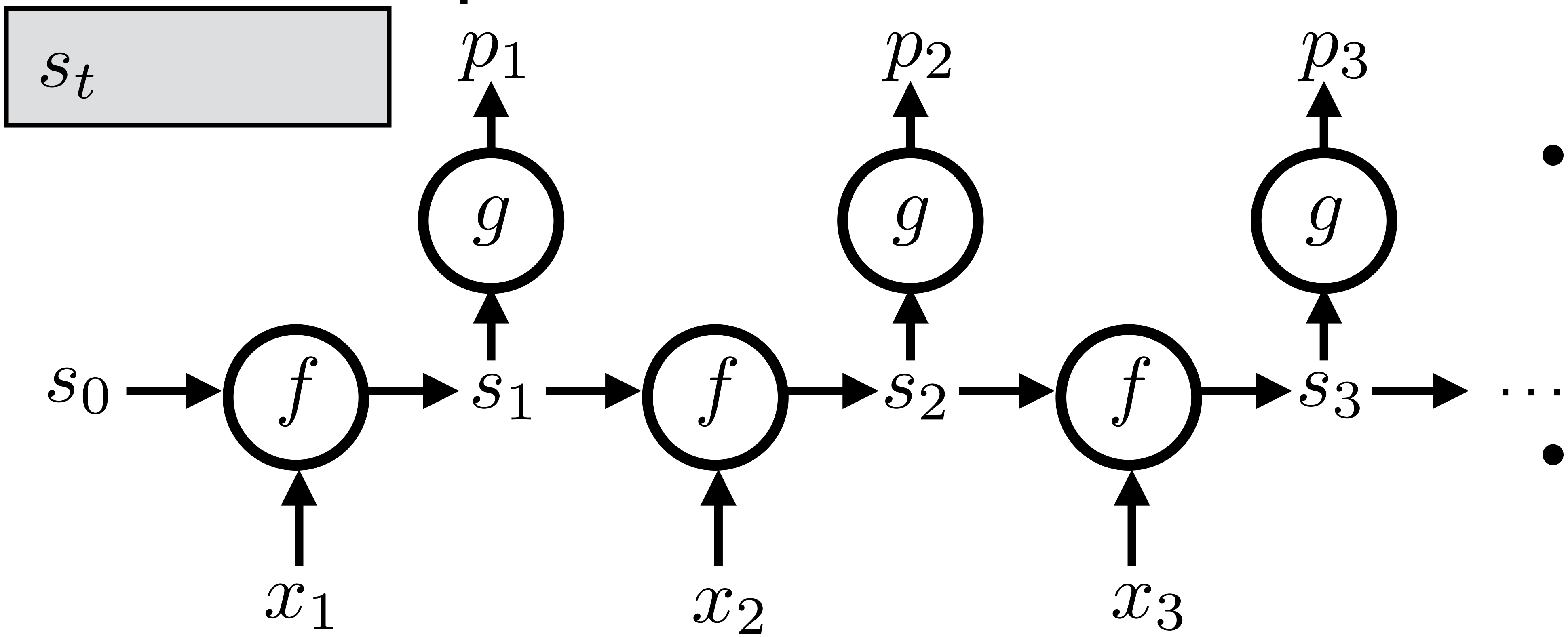
Can express as a state machine



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

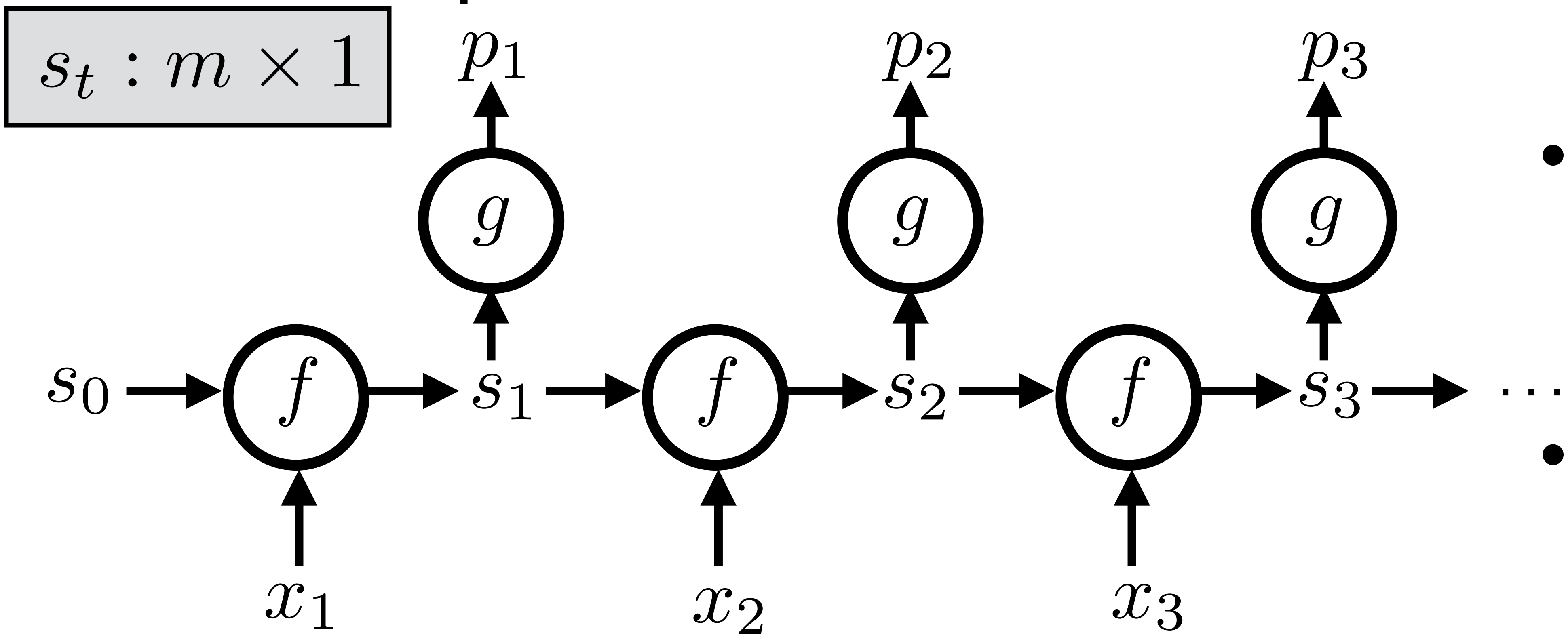
Can express as a state machine



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

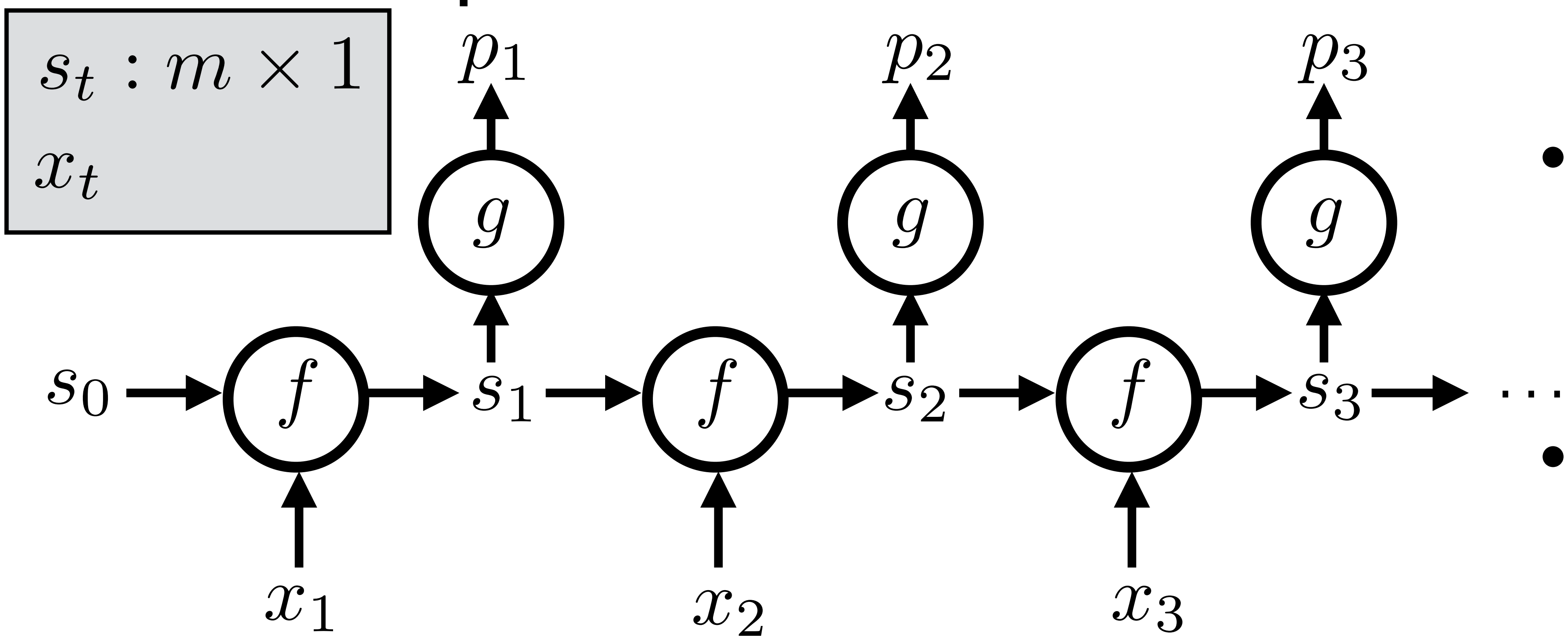
Can express as a state machine



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

Can express as a state machine

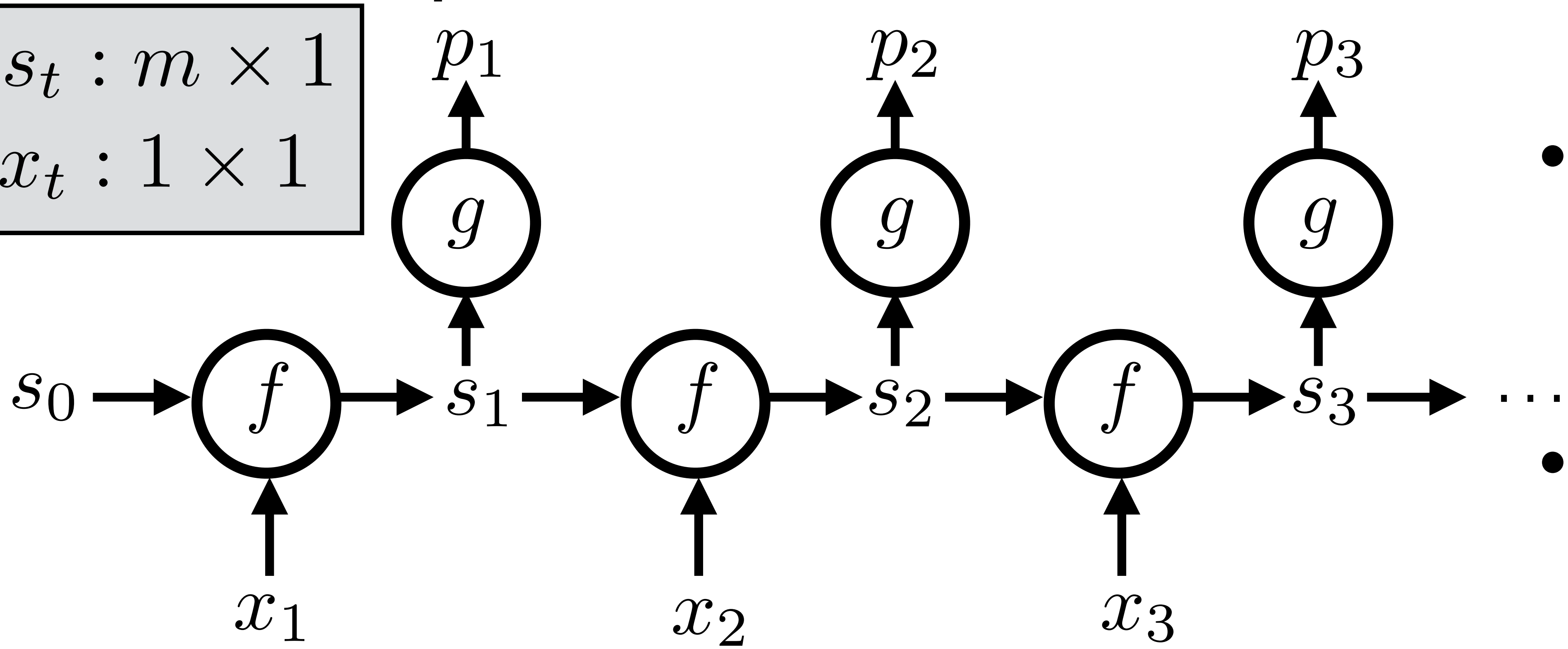


- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

Can express as a state machine

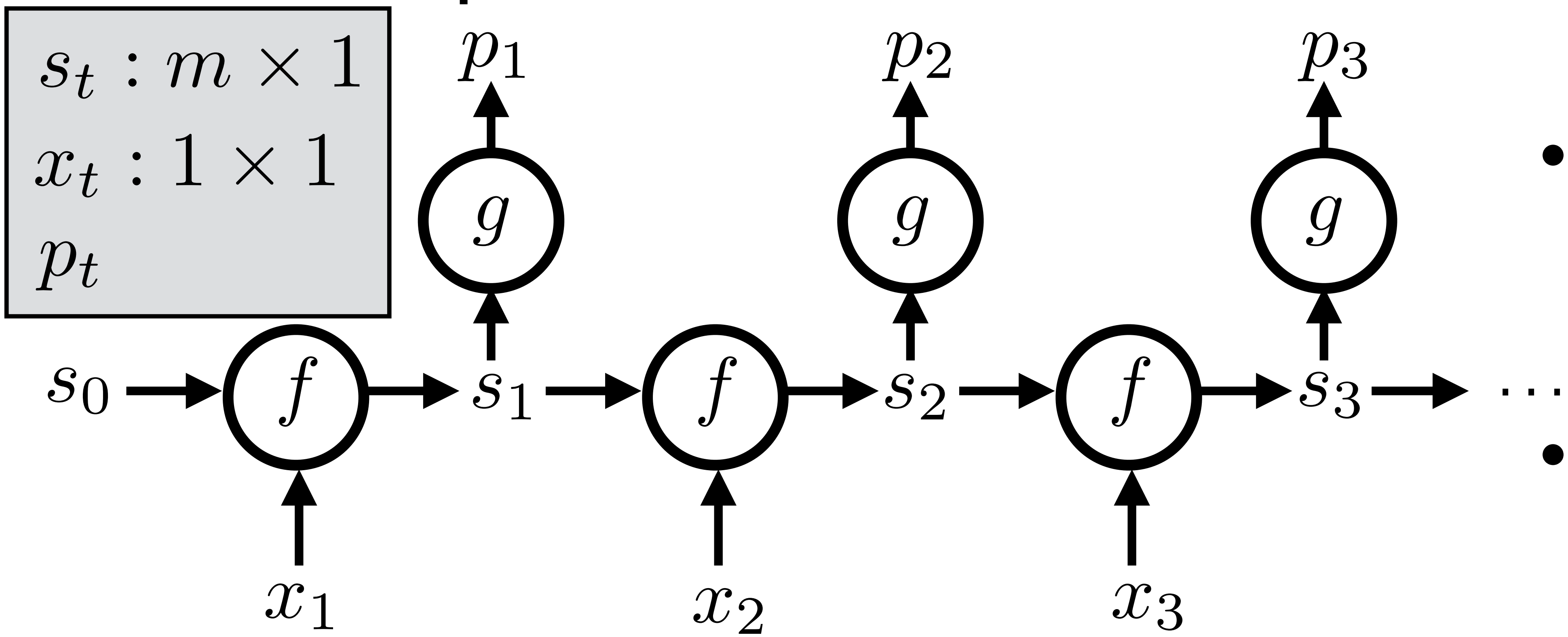
$s_t : m \times 1$
$x_t : 1 \times 1$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

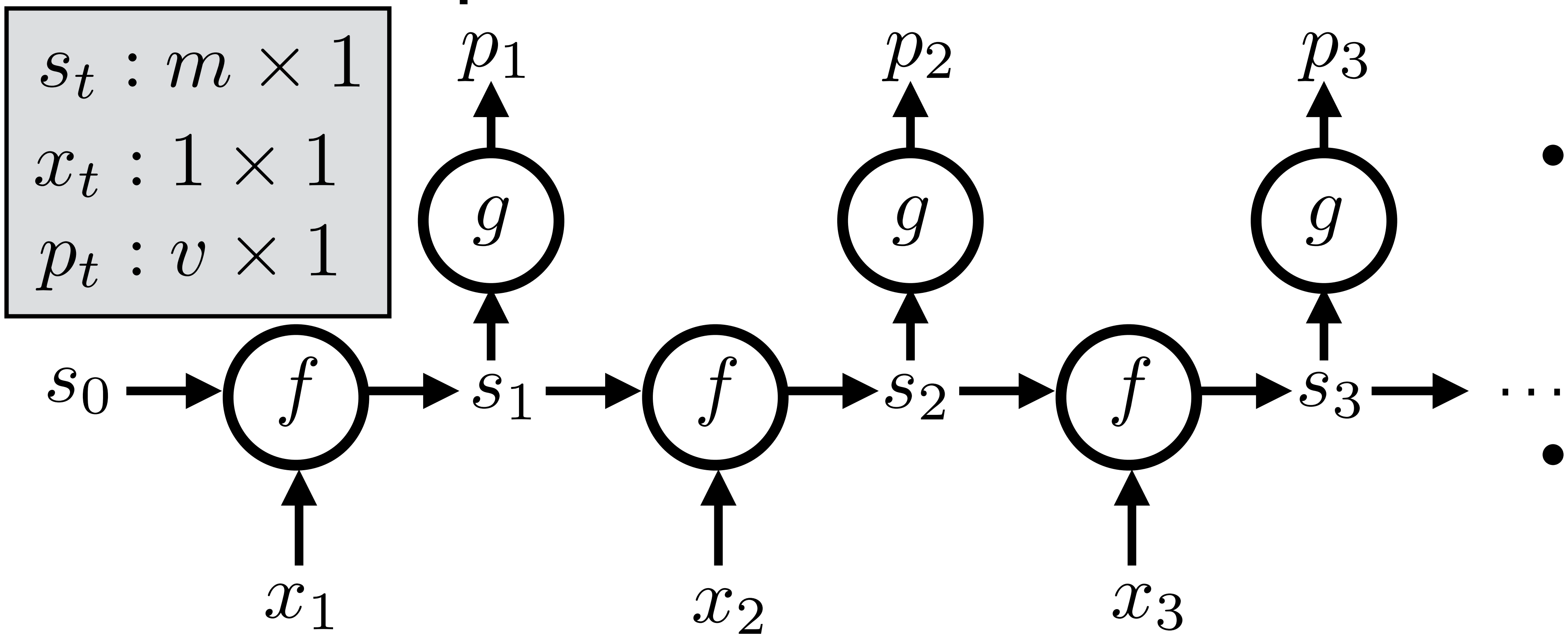
Can express as a state machine



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

Can express as a state machine



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

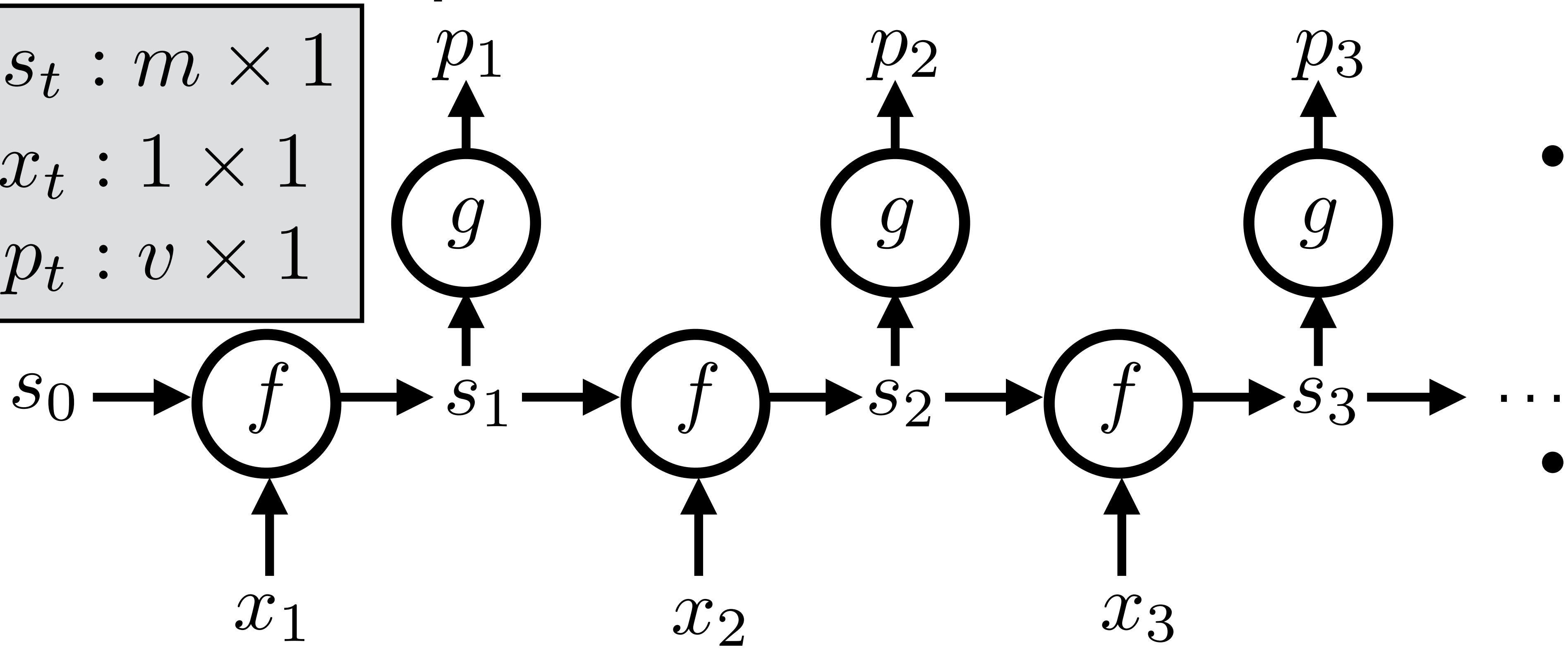
- Example: Alphabet {0,1}; state is last *m* = 3 characters

Can express as a state machine

$s_t : m \times 1$

$x_t : 1 \times 1$

$p_t : v \times 1$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1};
  state is last *m* = 3 characters

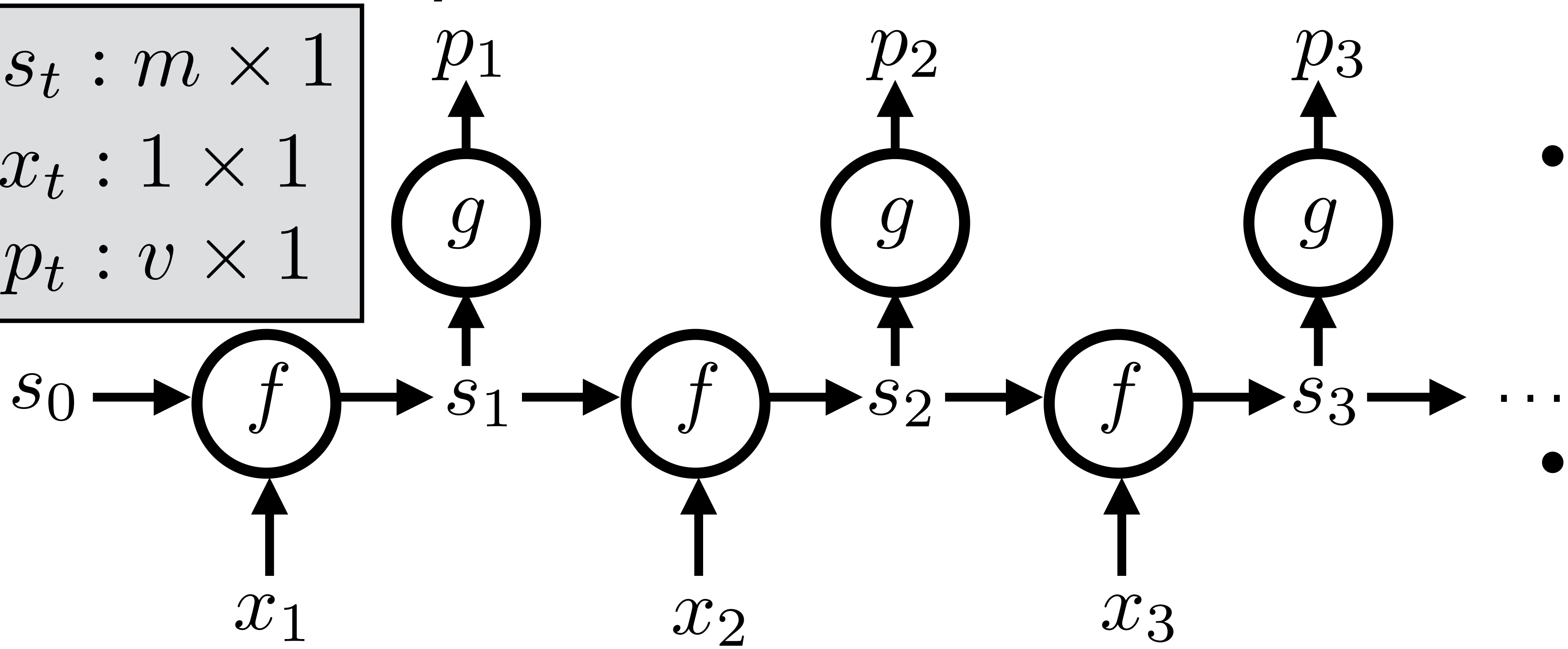
$$s_t = f(s_{t-1}, x_t) =$$

Can express as a state machine

$s_t : m \times 1$

$x_t : 1 \times 1$

$p_t : v \times 1$

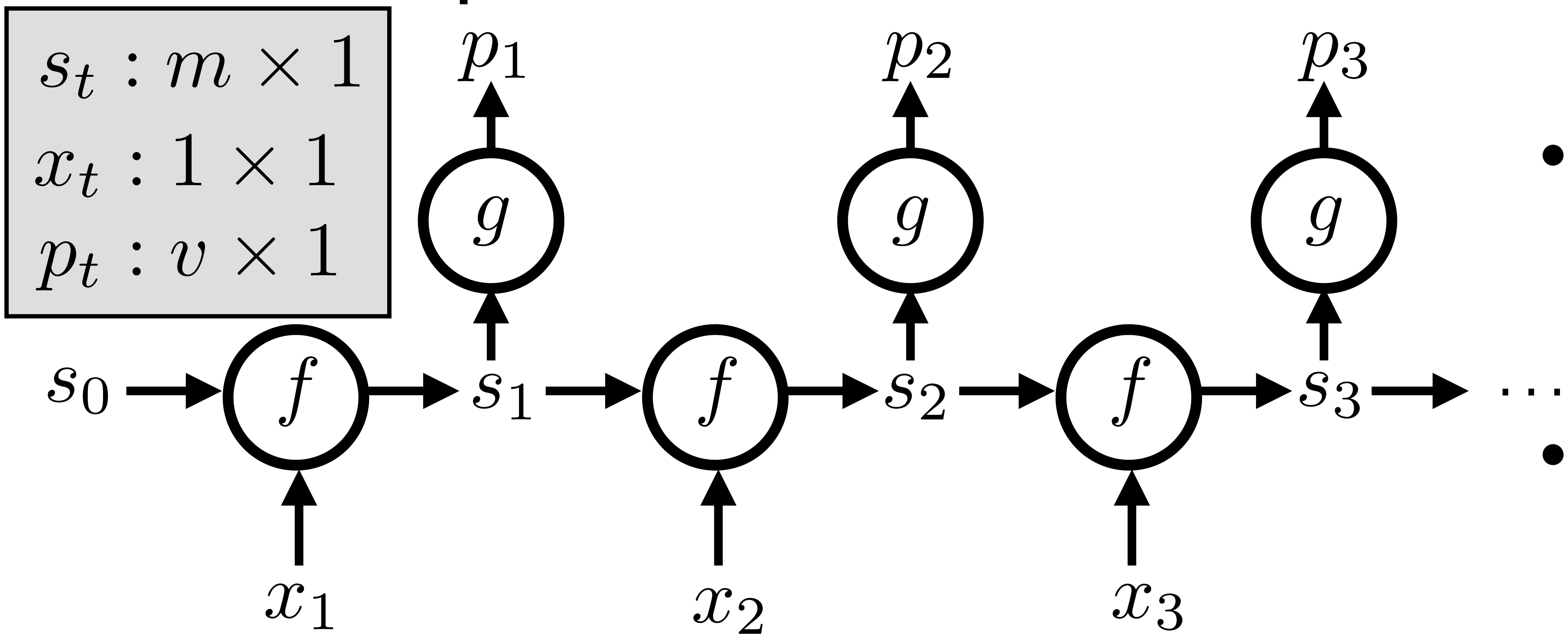


- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) =$$

Can express as a state machine



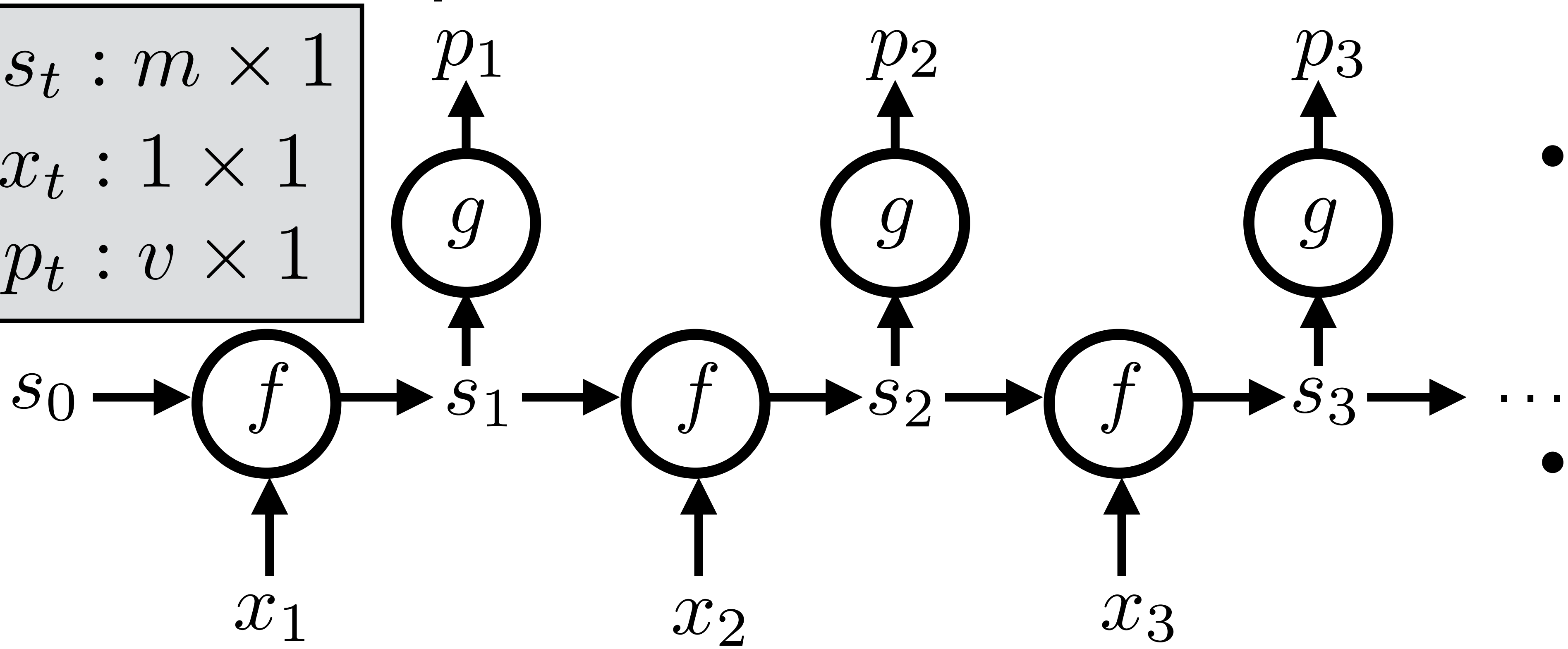
- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \boxed{?}\; x_t + \boxed{?}\; s_{t-1}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



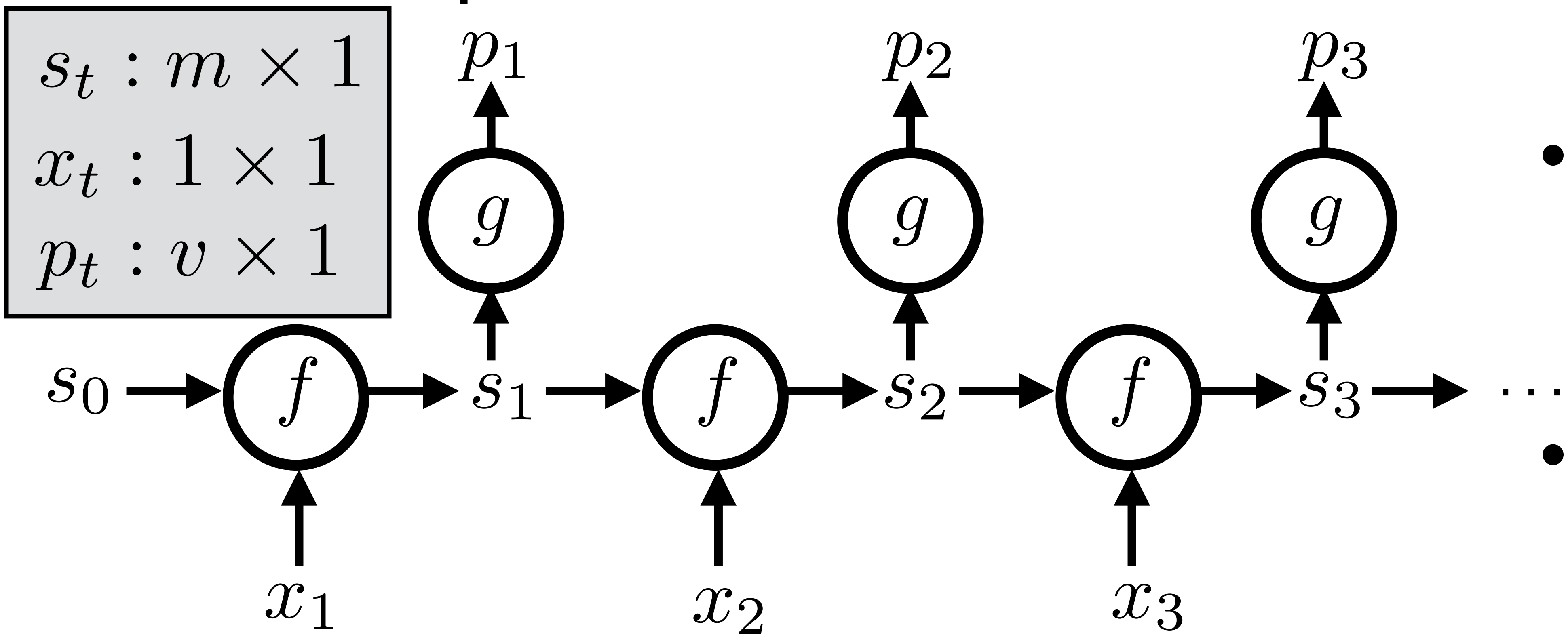
- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \boxed{?}\, x_t + \boxed{?}\, s_{t-1}$$

3 x 1

Can express as a state machine



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

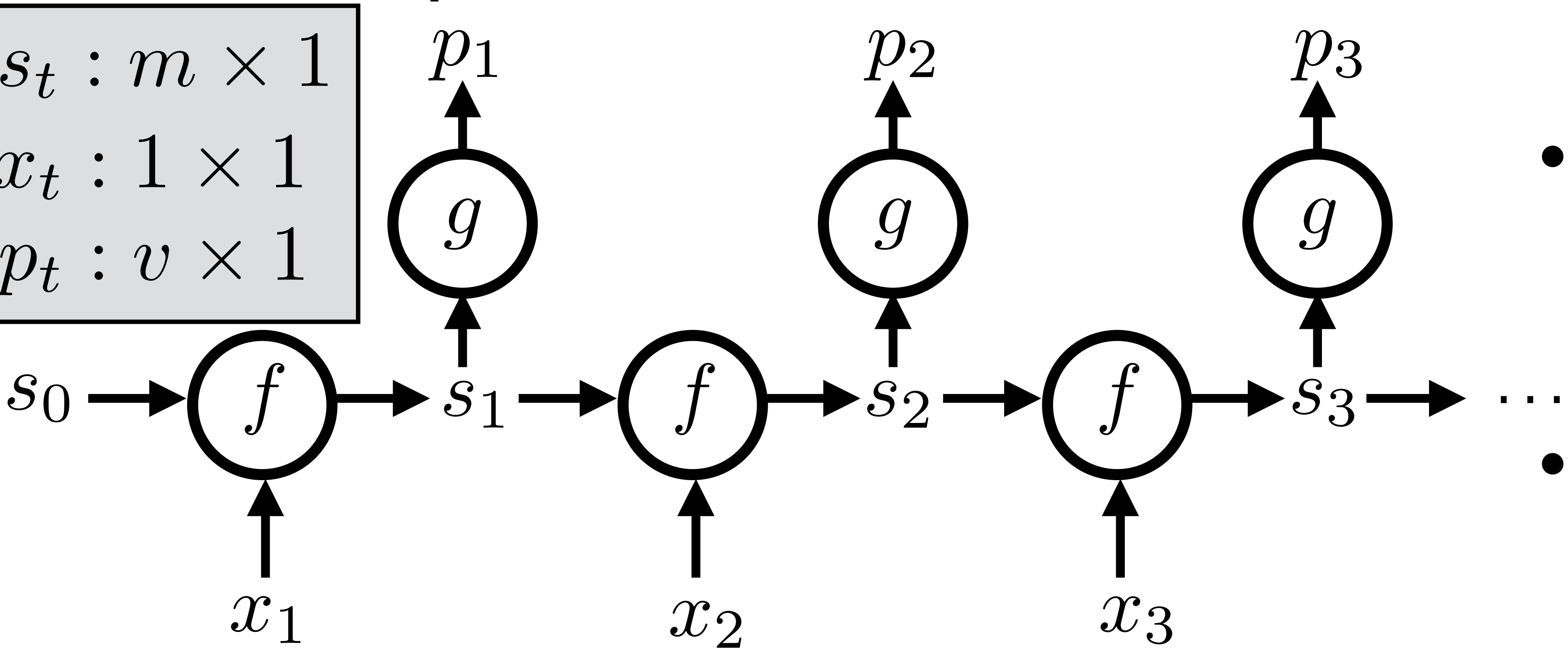
$$s_t = f(s_{t-1}, x_t) = \boxed{?}\; x_t + \boxed{?}\; s_{t-1}$$

3 x 1

1 x 1

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

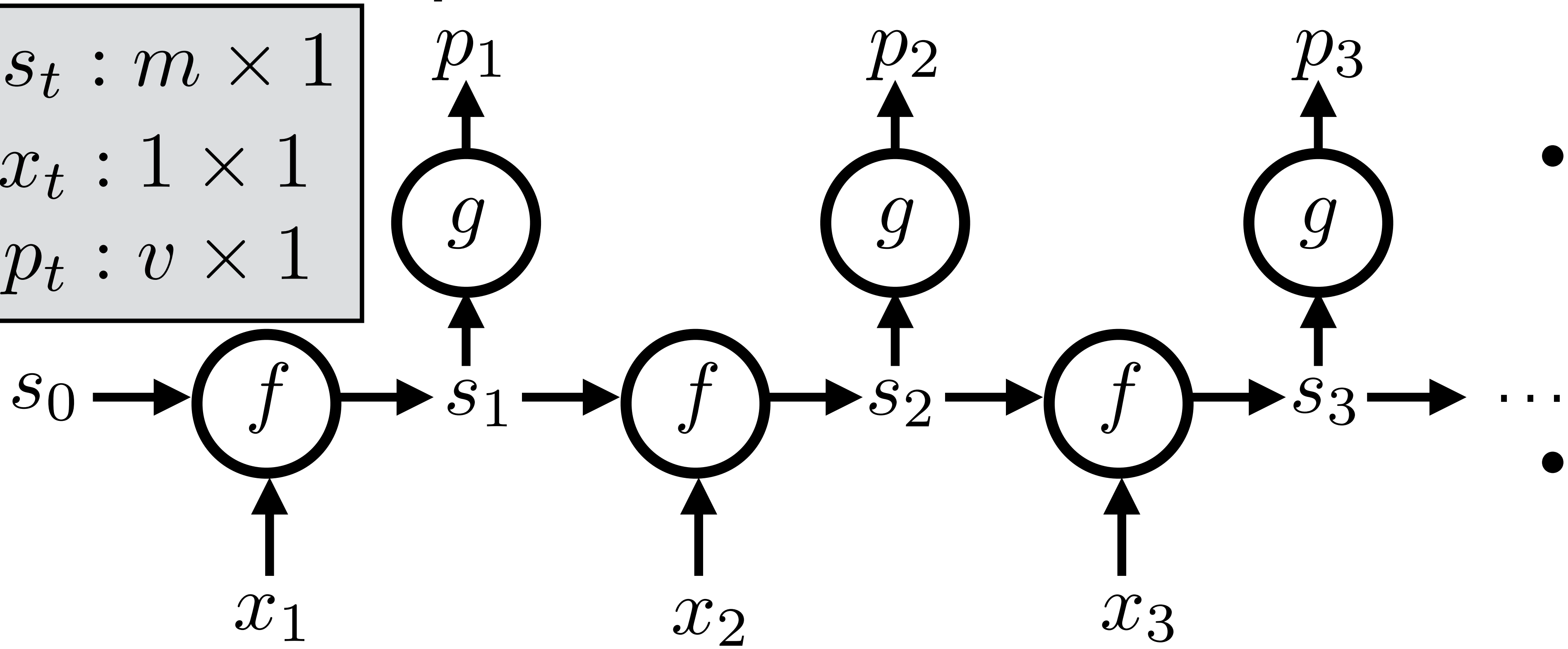
$$s_t = f(s_{t-1}, x_t) = \boxed{?}\, x_t + \boxed{?}\, s_{t-1}$$

3 x 1

3 x 1, 1 x 1

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

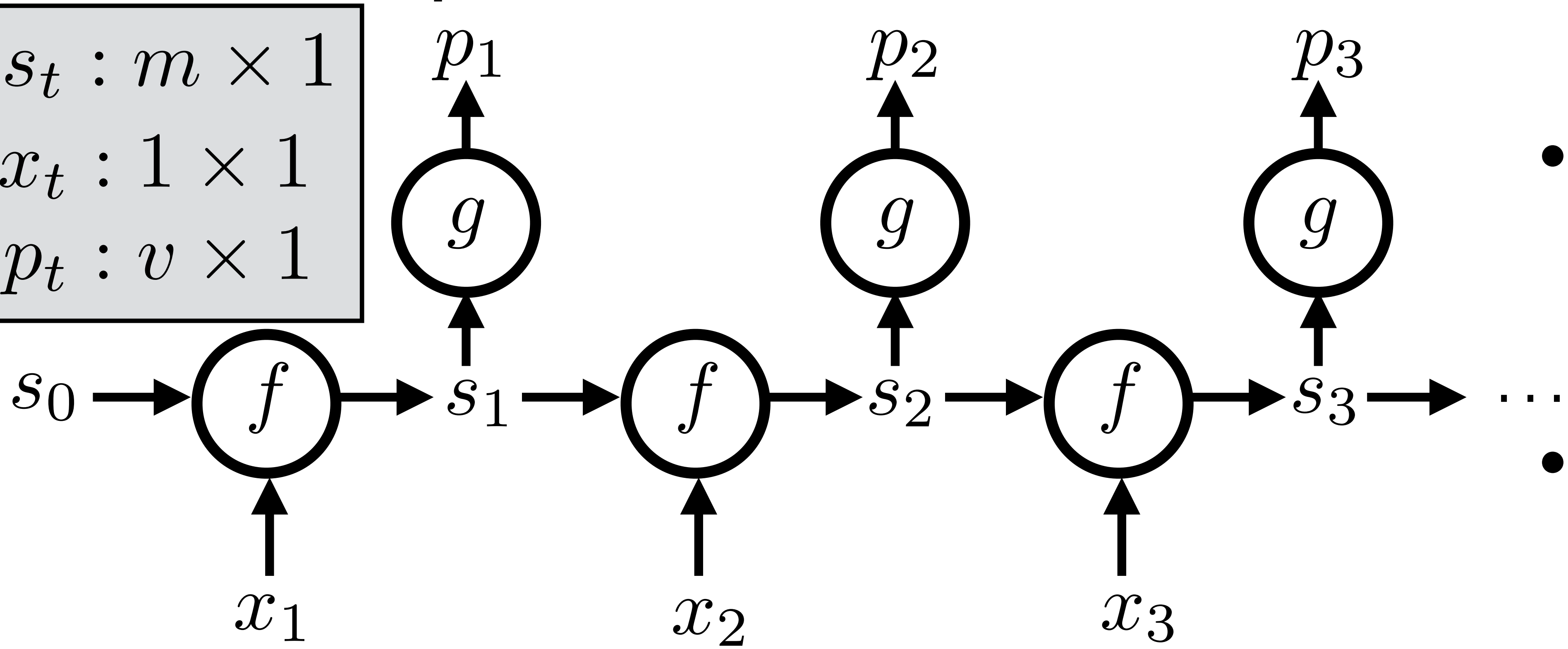
$$s_t = f(s_{t-1}, x_t) = \boxed{?}\; x_t + \boxed{?}\; s_{t-1}$$

3 x 1

3 x 1, 1 x 1

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

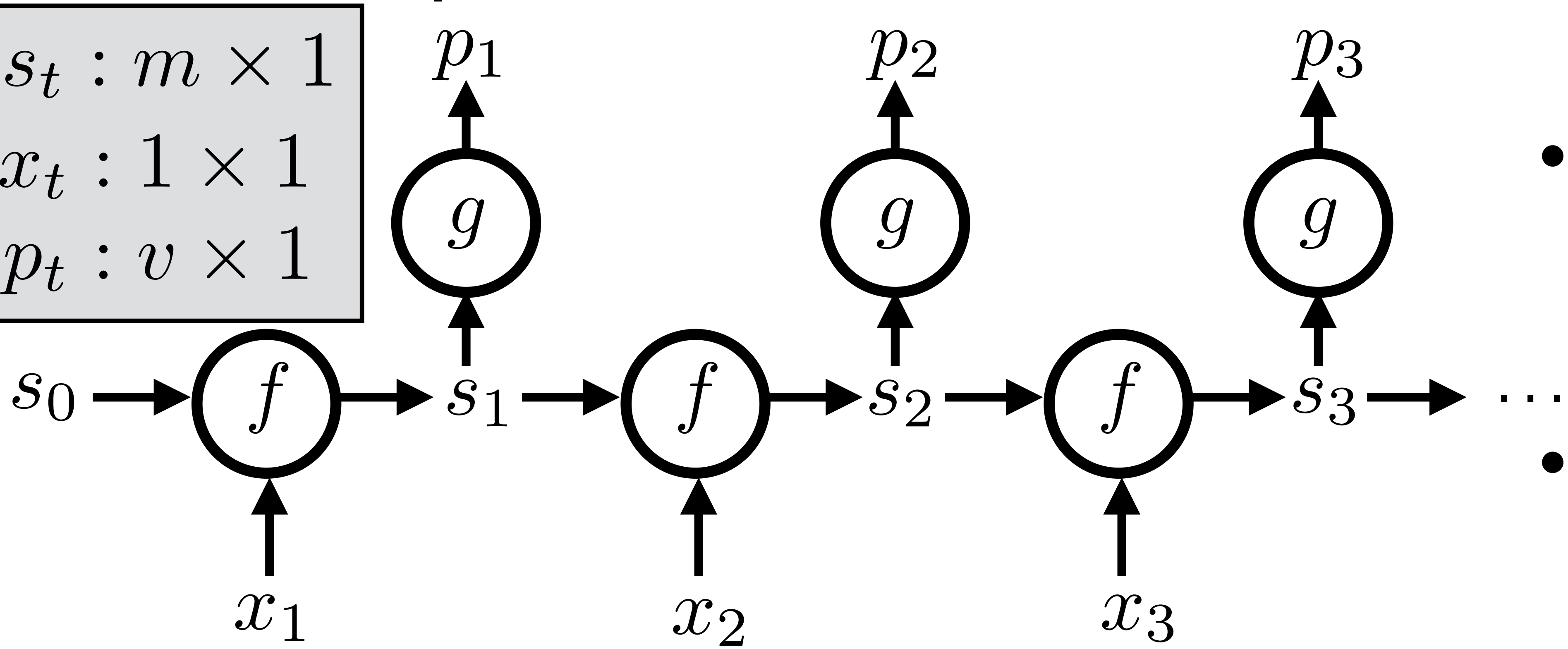
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \boxed{?} \; s_{t-1}$$

Can express as a state machine

$s_t : m \times 1$

$x_t : 1 \times 1$

$p_t : v \times 1$

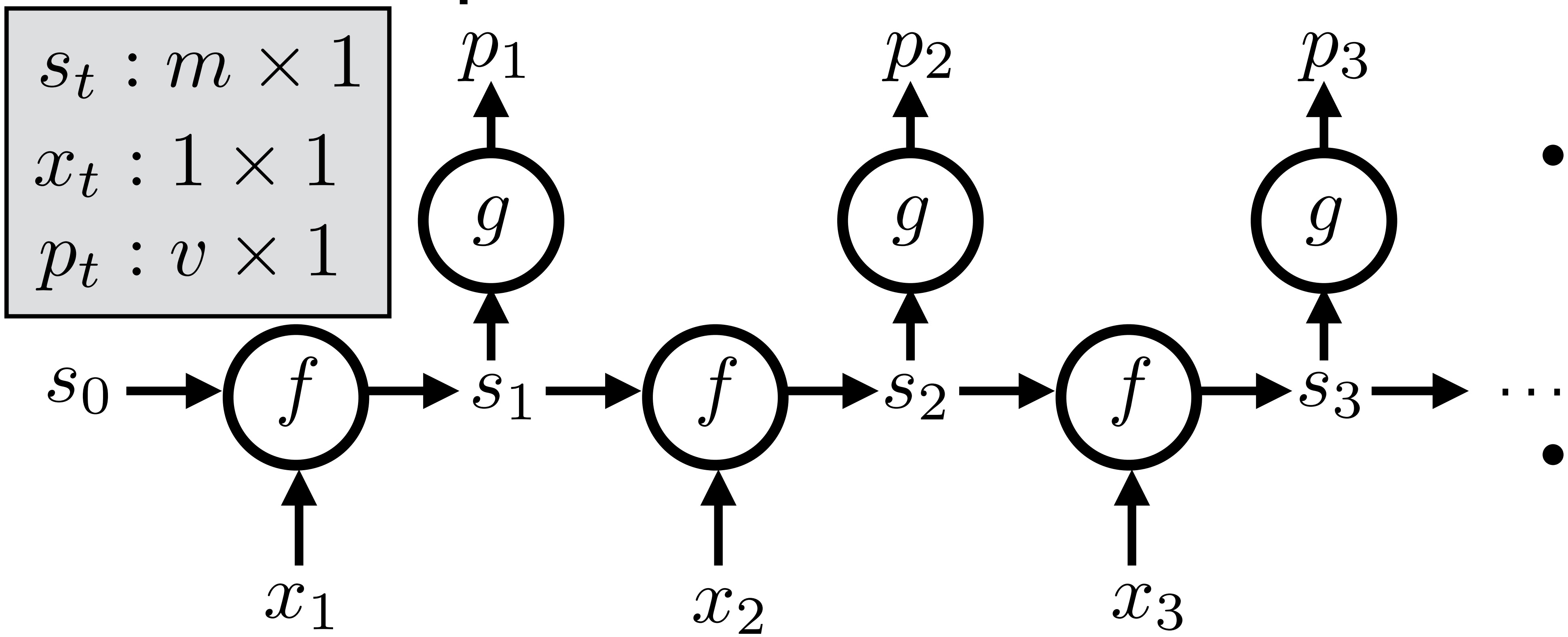


- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last m = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \boxed{?} \; s_{t-1}$$

Can express as a state machine

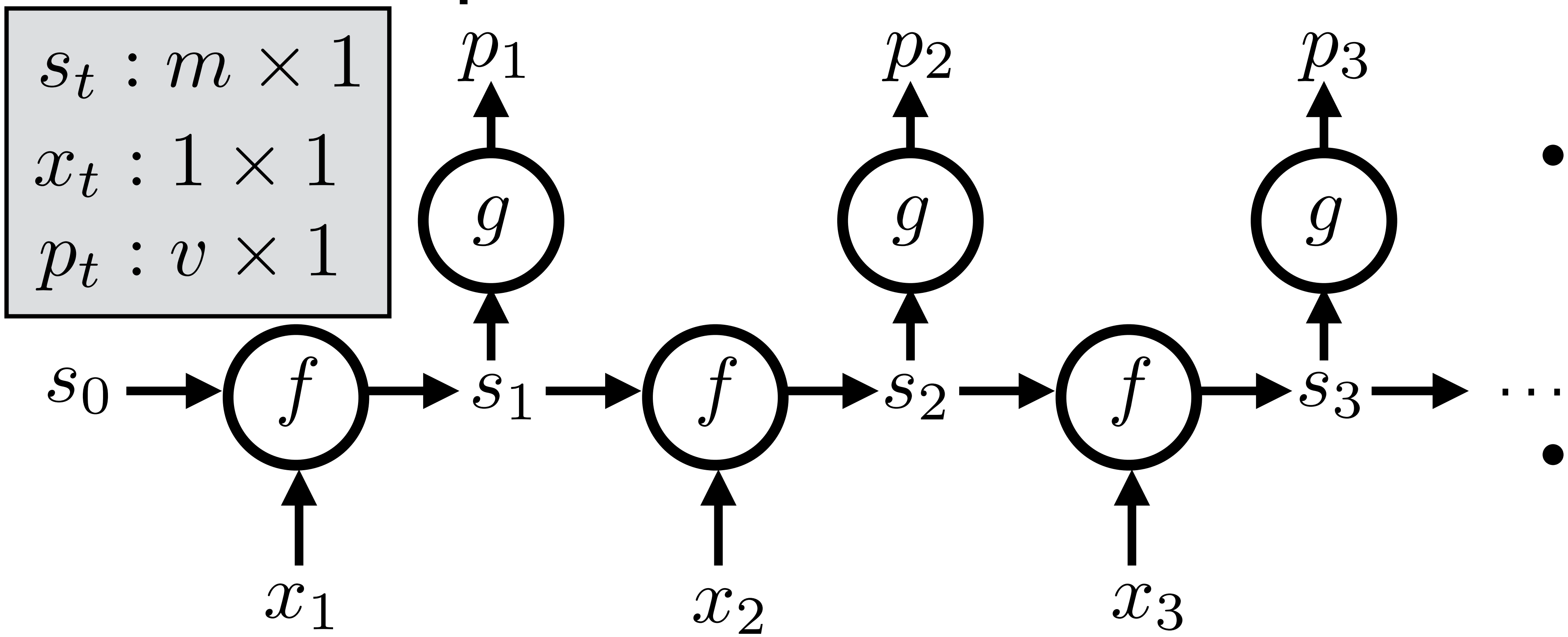


- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

Can express as a state machine

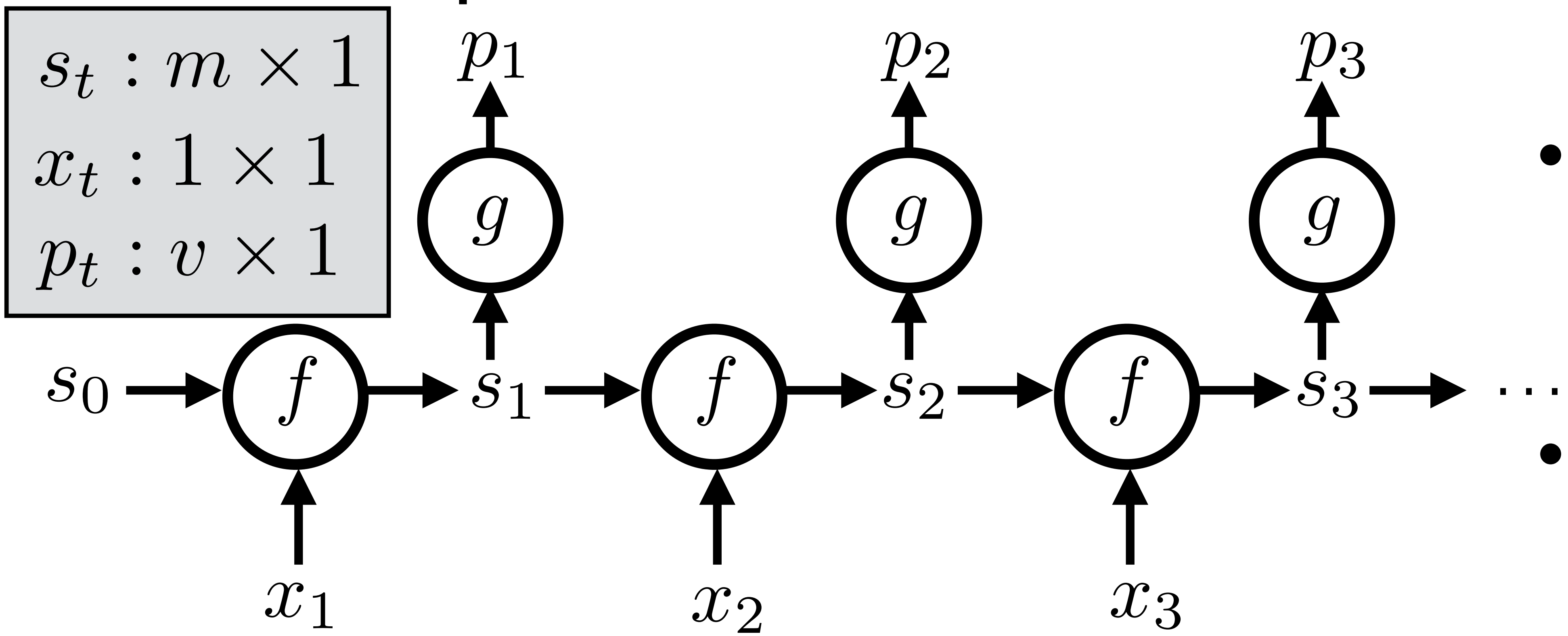


- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

Can express as a state machine

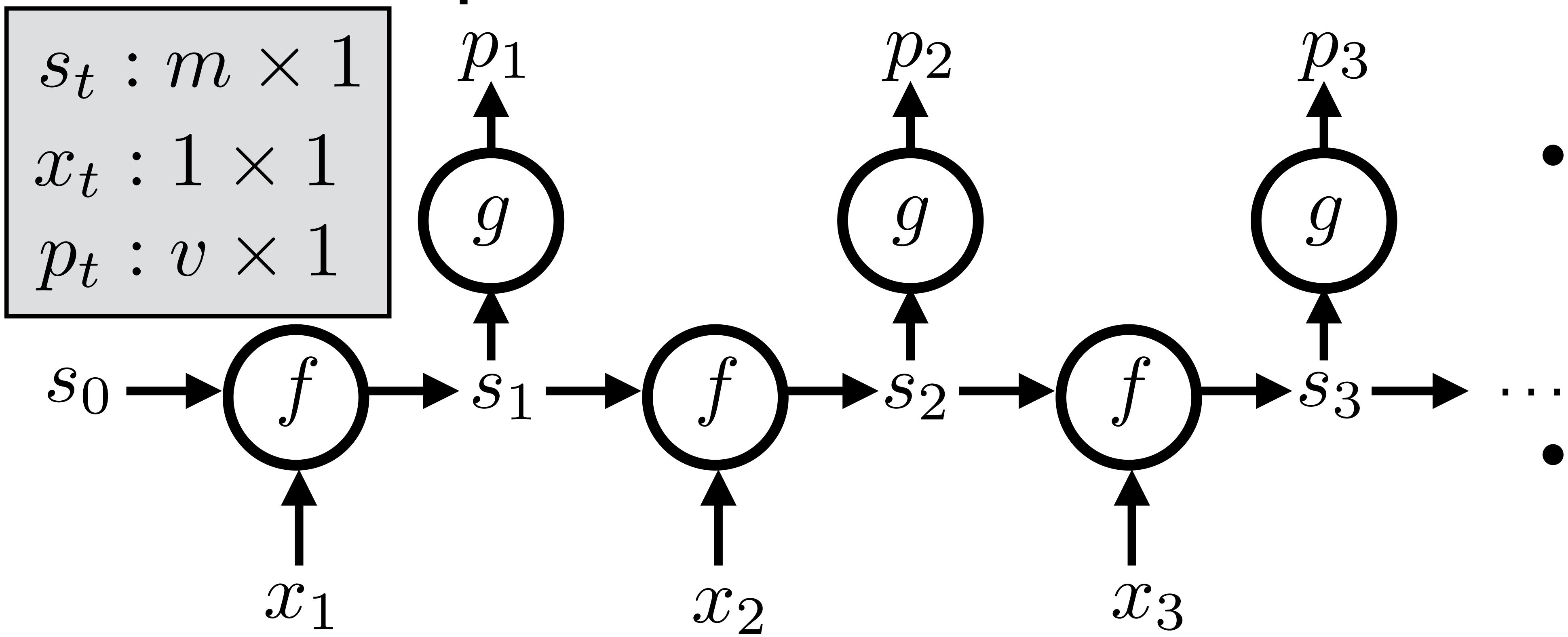


- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1};
  state is last $m = 3$ characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

Can express as a state machine

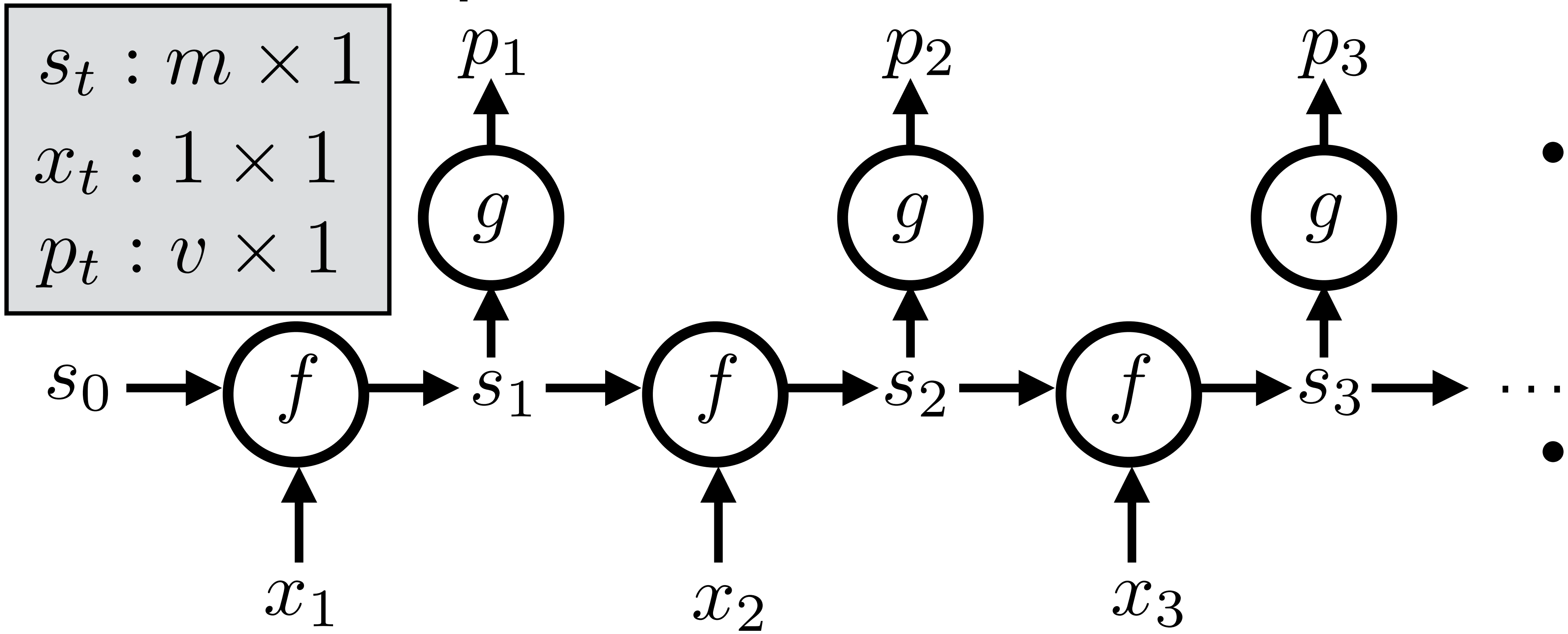


- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

Can express as a state machine



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

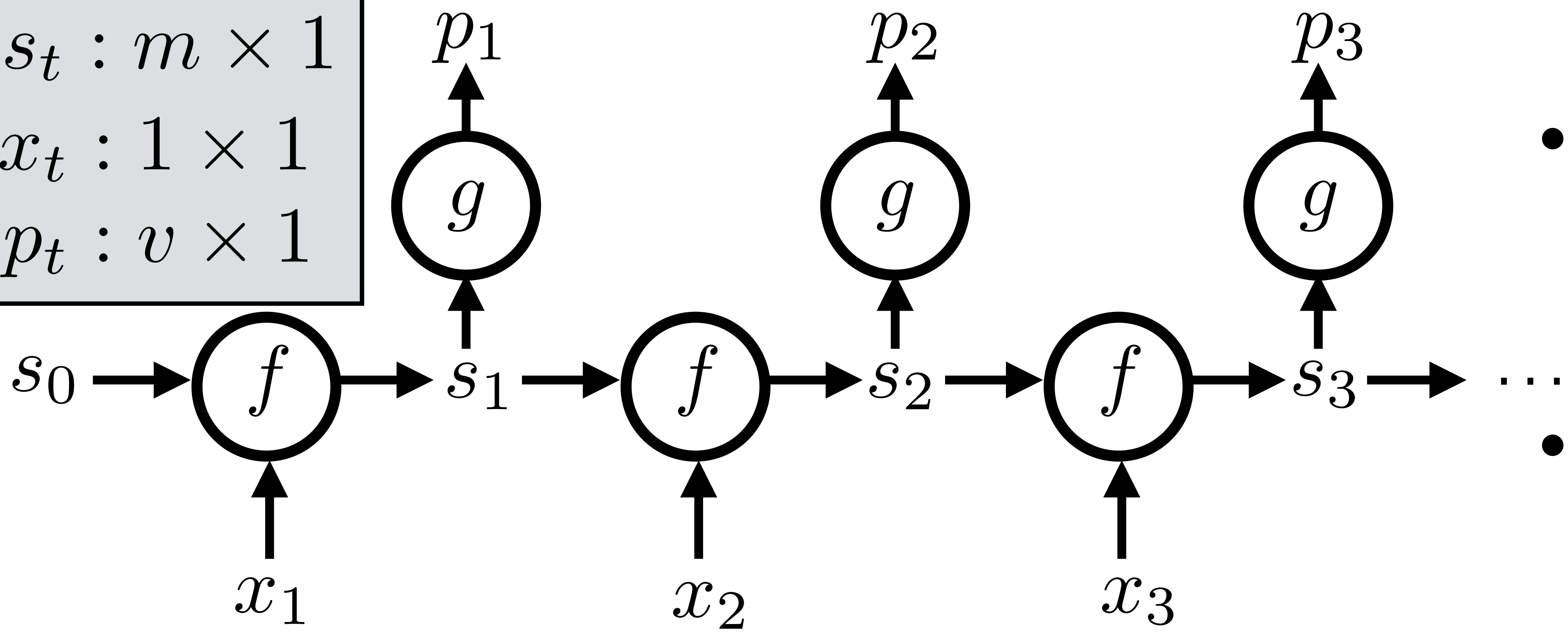
- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$p_t = g(s_t)$$
$$=$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

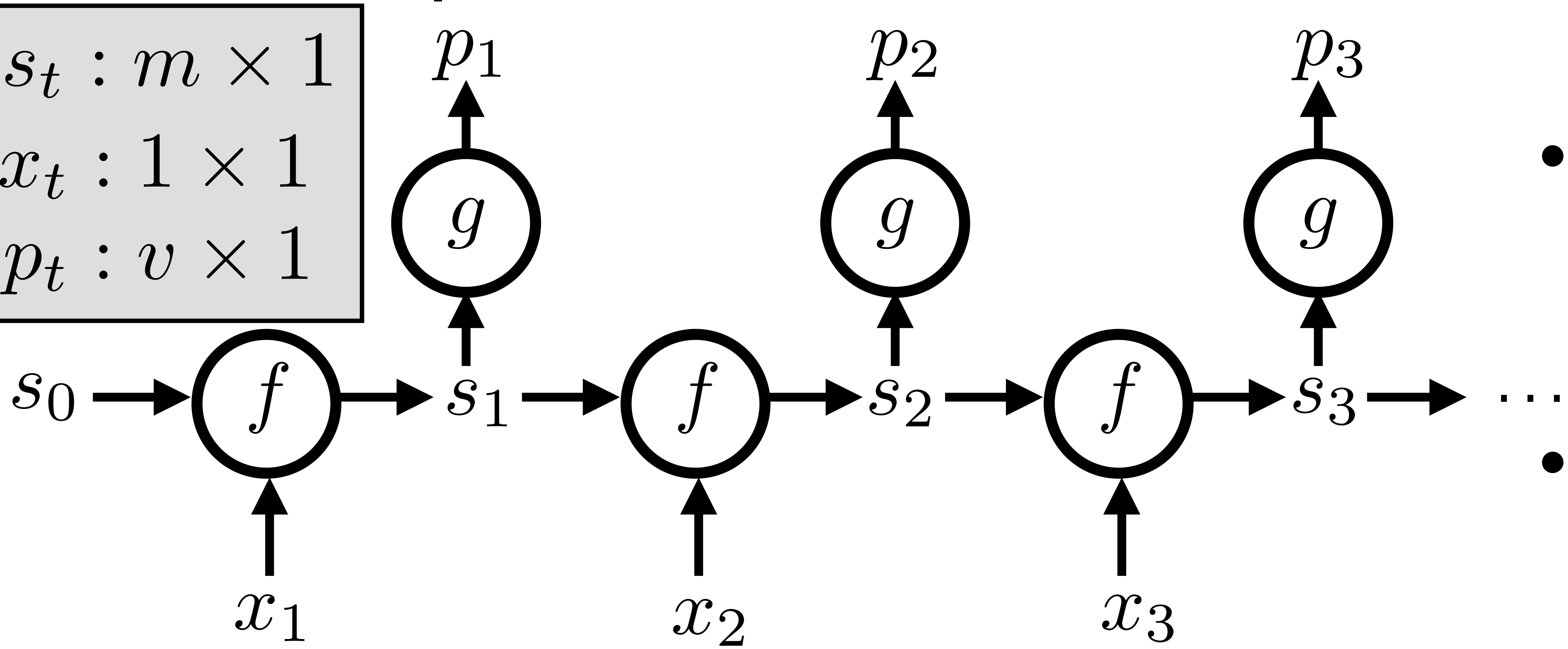
- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

Can express as a state machine

$$s_t : m \times 1$$
$$x_t : 1 \times 1$$
$$p_t : v \times 1$$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

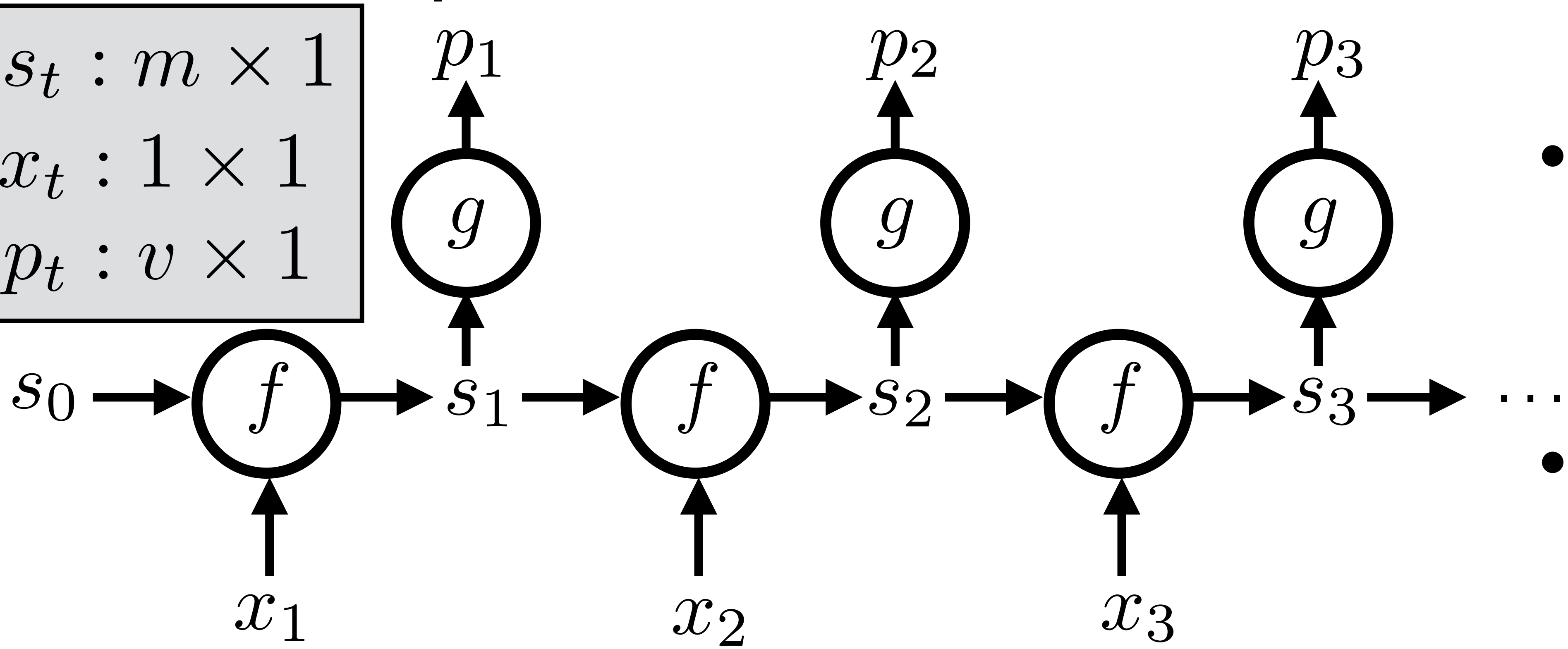
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

2-class logistic regression

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

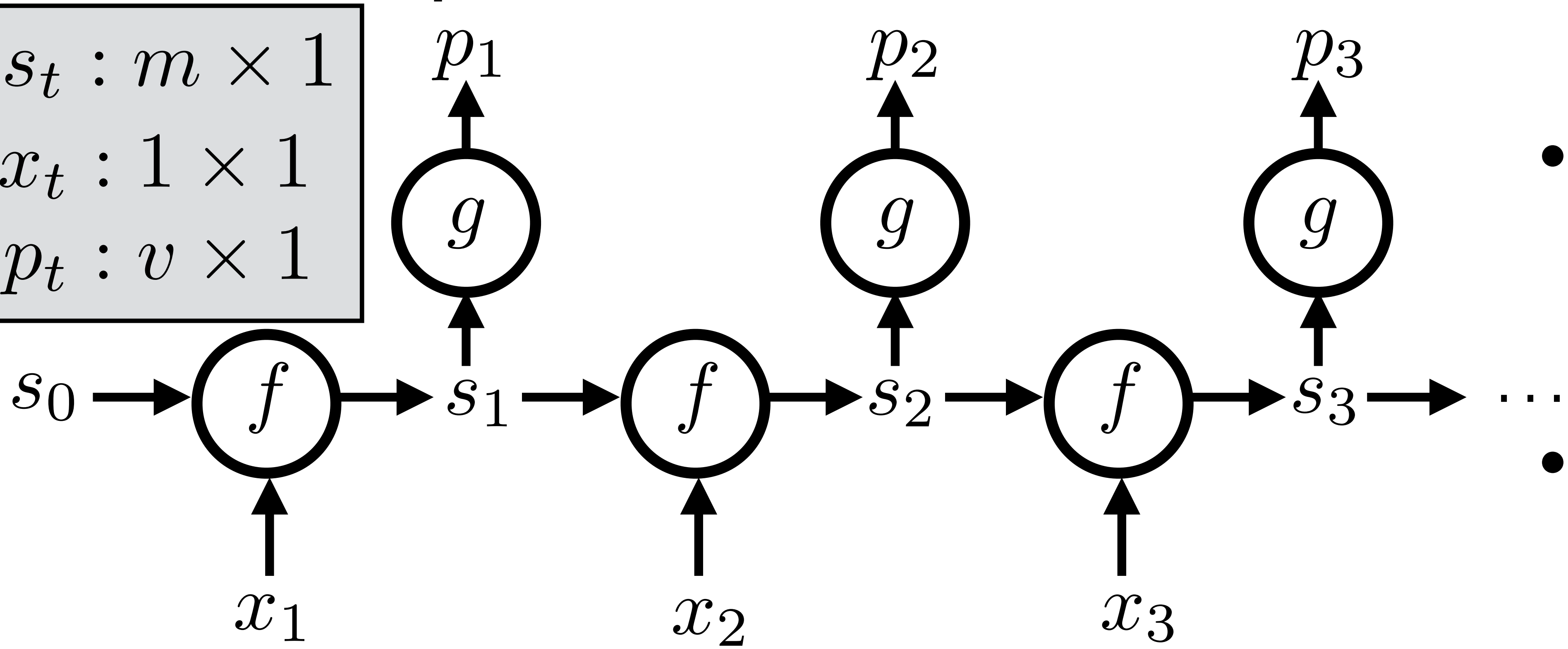
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{1 \times 3} s_t + W^o_0) \end{aligned}$$

2-class logistic regression

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$p_t = g(s_t)$$
$$= f_2(W^o s_t + W_0^o)$$

1 x 3   1 x 1

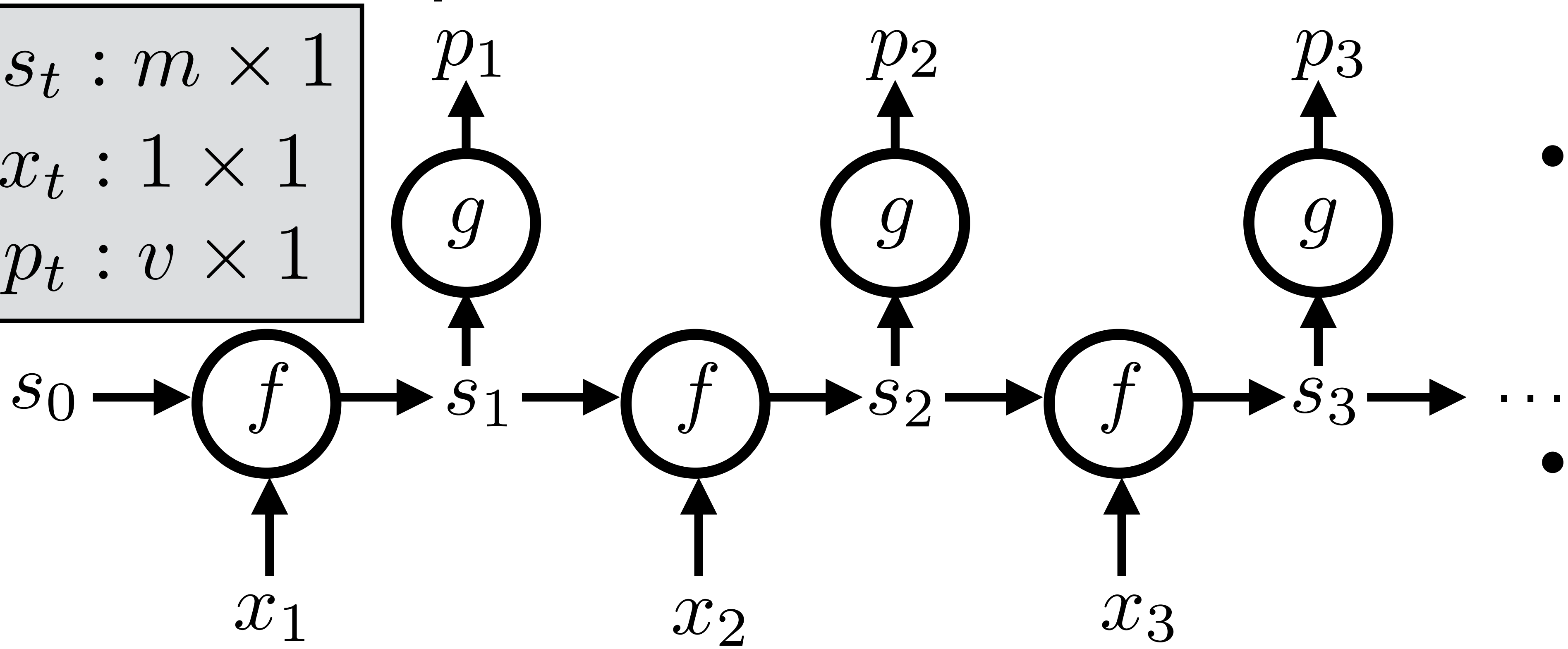
2-class logistic regression

Can express as a state machine

$s_t : m \times 1$

$x_t : 1 \times 1$

$p_t : v \times 1$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

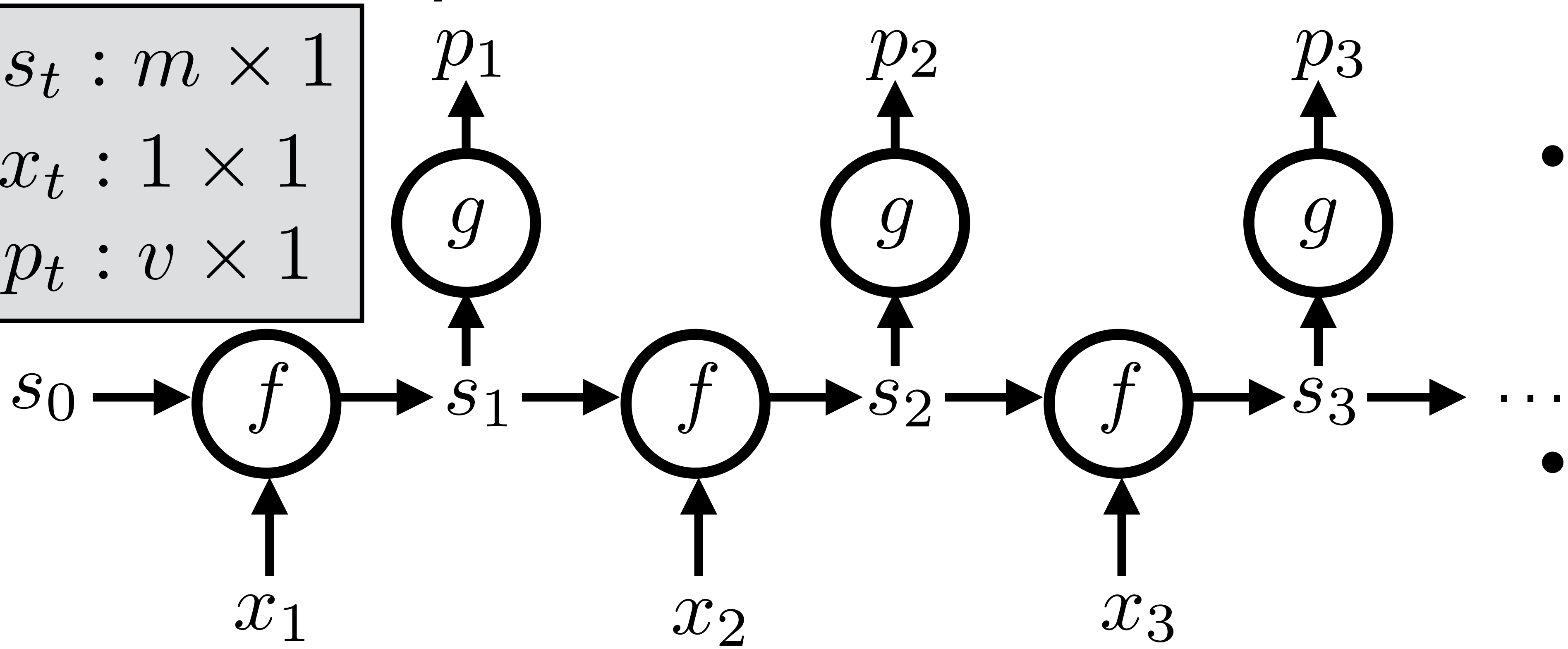
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{1 \times 3} s_t + \underbrace{W^o_0}_{1 \times 1}) \end{aligned}$$

2-class logistic regression

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

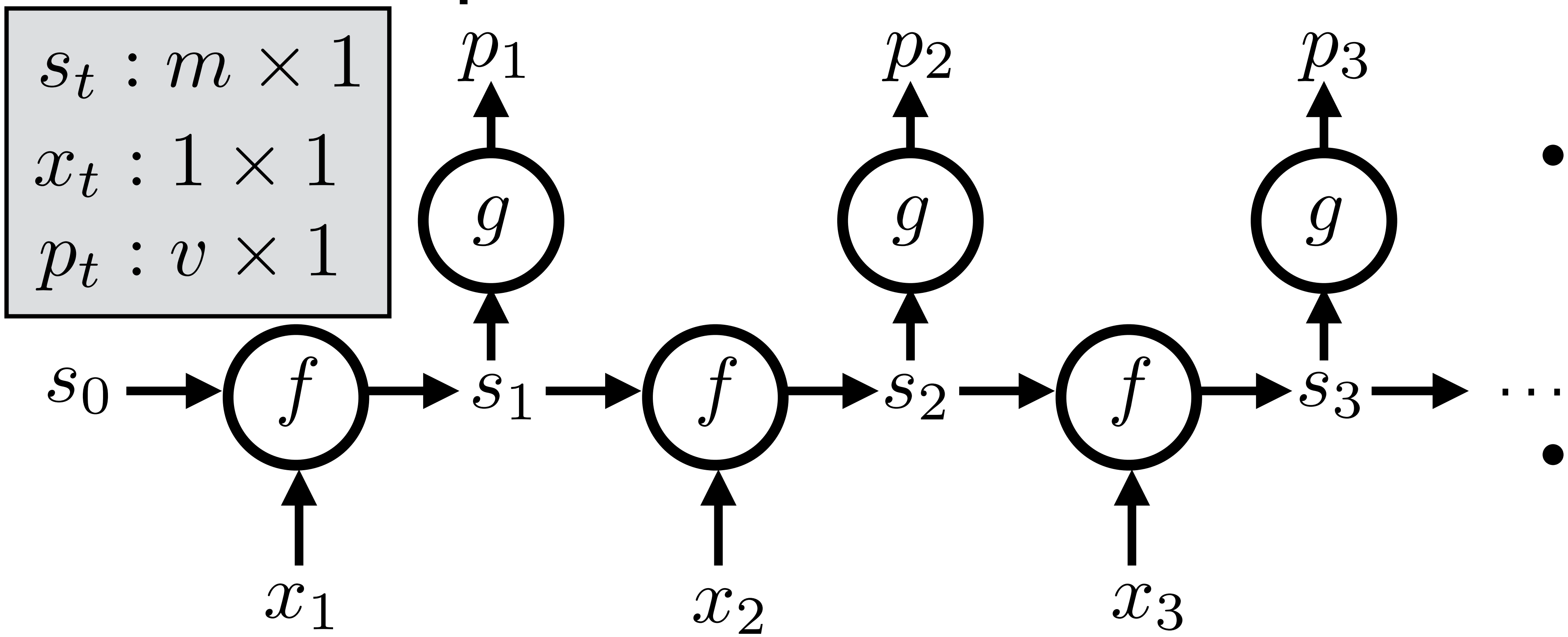
- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{1 \times 3} s_t + \underbrace{W_0^o}_{1 \times 1}) \end{aligned}$$

v-class logistic regression

Can express as a state machine



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

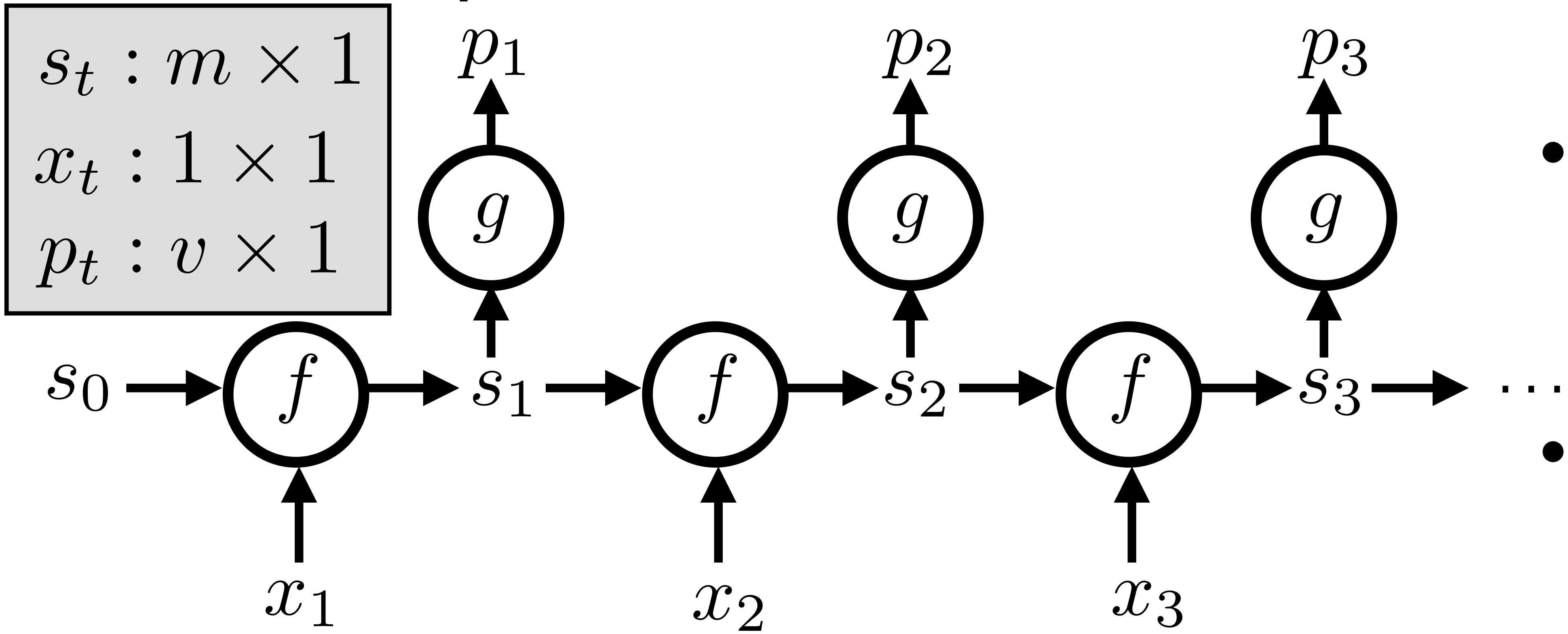
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$p_t = g(s_t) = f_2(W^o s_t + W^o_0)$$

1 x 3   1 x 1

v-class logistic regression

Can express as a state machine



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

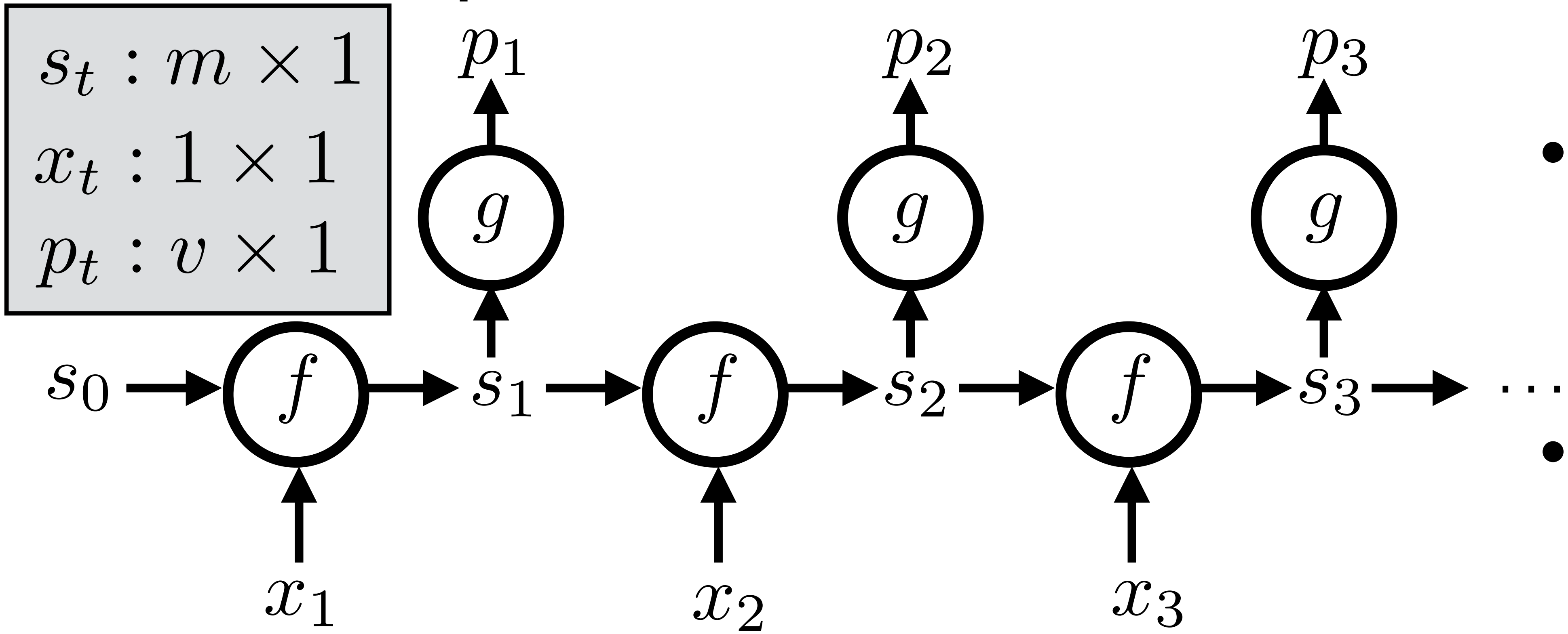
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

v x 3   *v* x 1

v-class logistic regression

Can express as a state machine



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

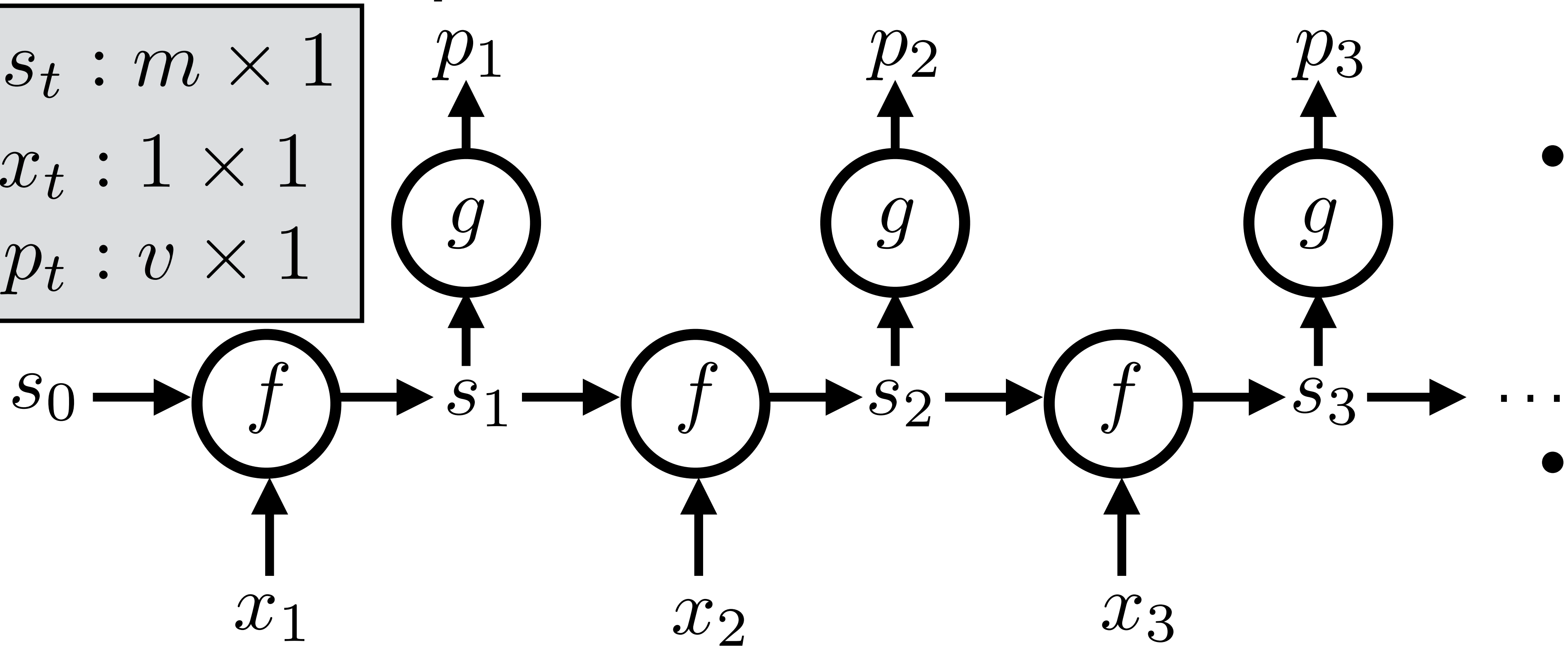
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times 3} s_t + \underbrace{W^o_0}_{v \times 1}) \end{aligned}$$

v-class logistic regression

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

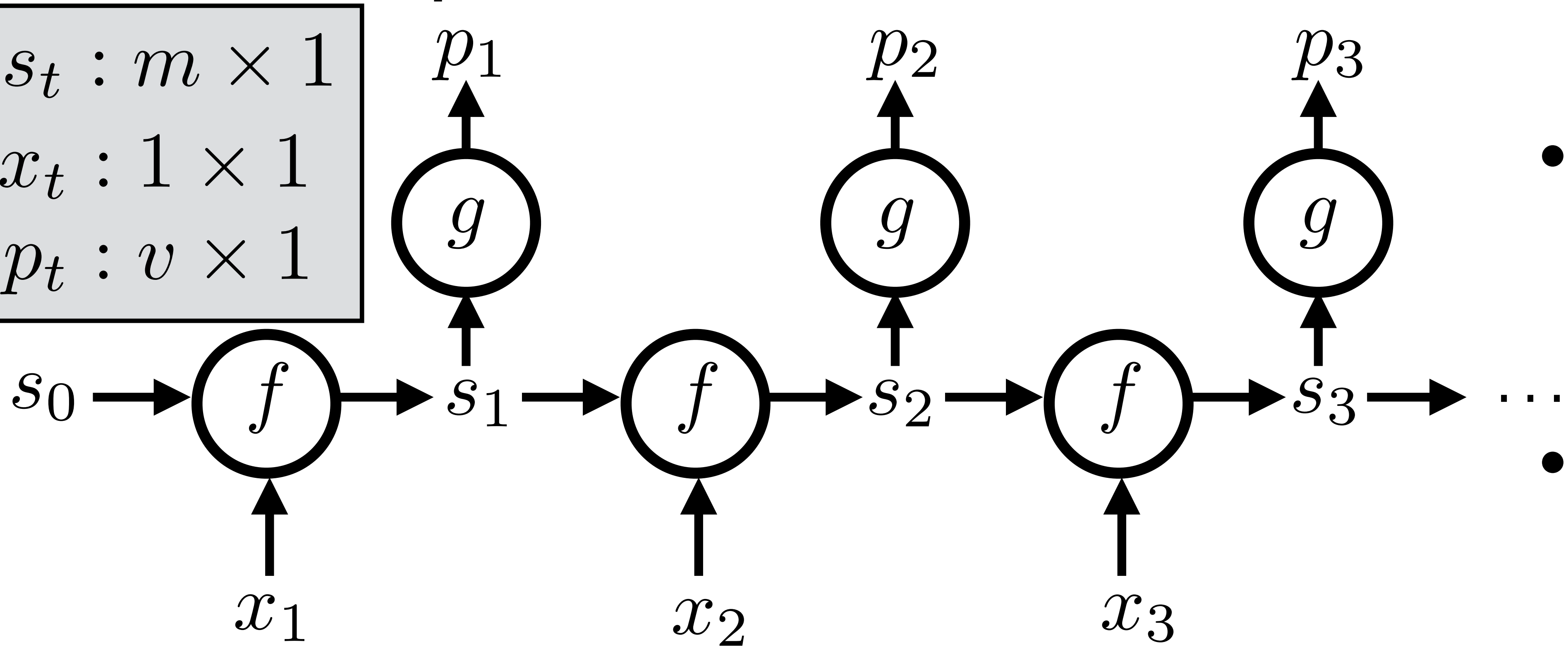
$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W^o_0) \end{aligned}$$

v x 3 *v* x 1

v-class logistic regression

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

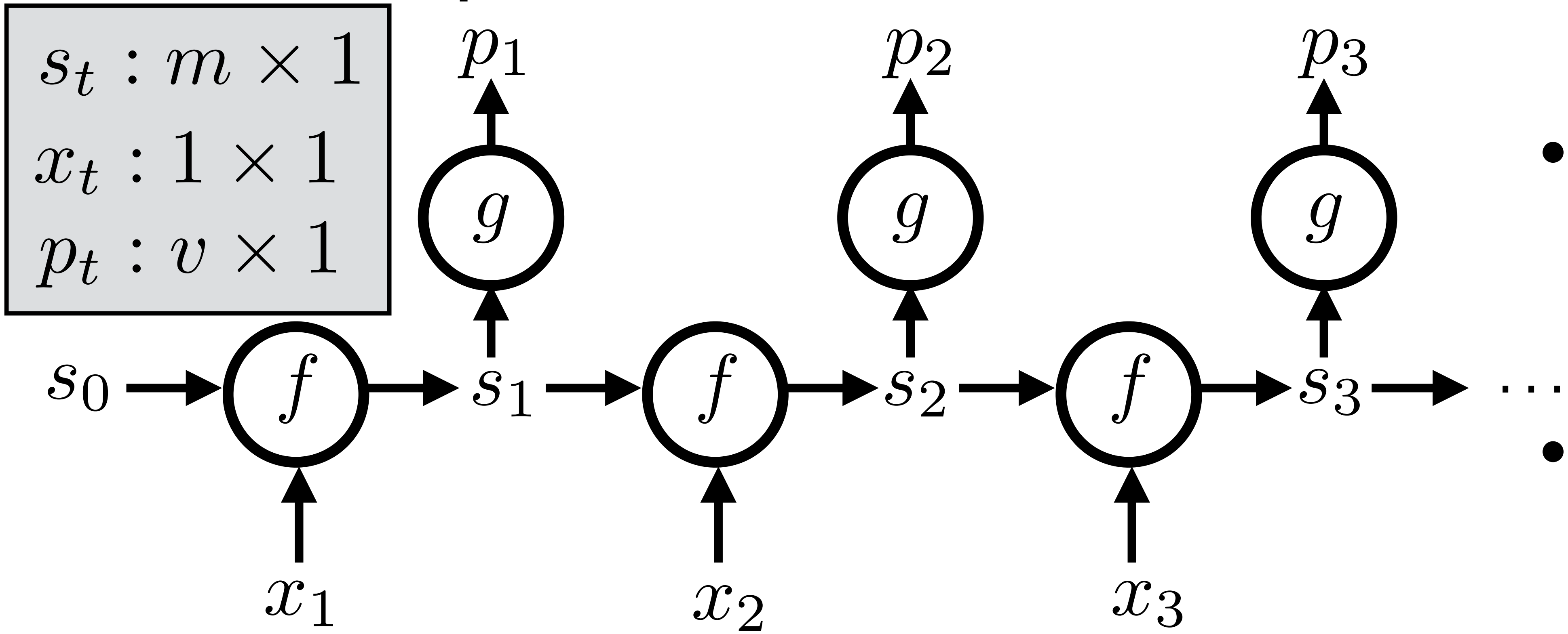
- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times 3} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

v-class logistic regression

Can express as a state machine



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

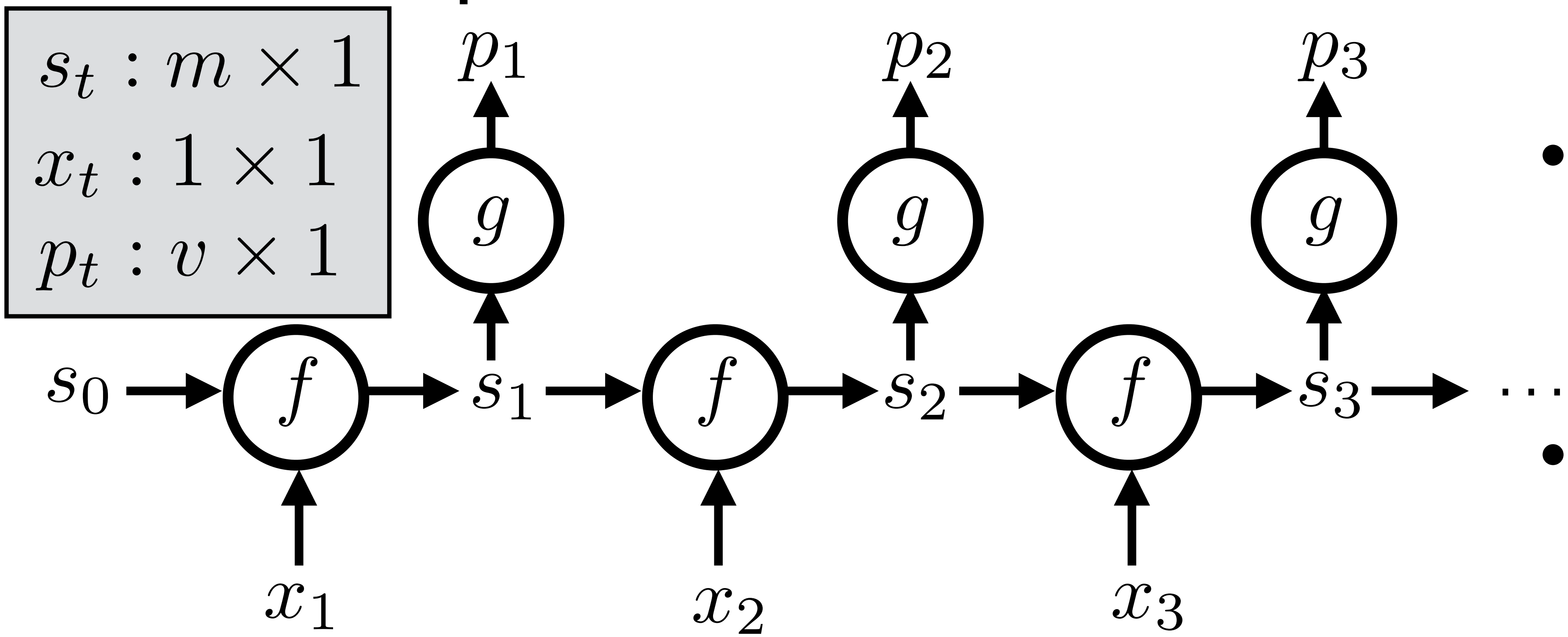
- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W^o_0}_{v \times 1}) \end{aligned}$$

v-class logistic regression

Can express as a state machine



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

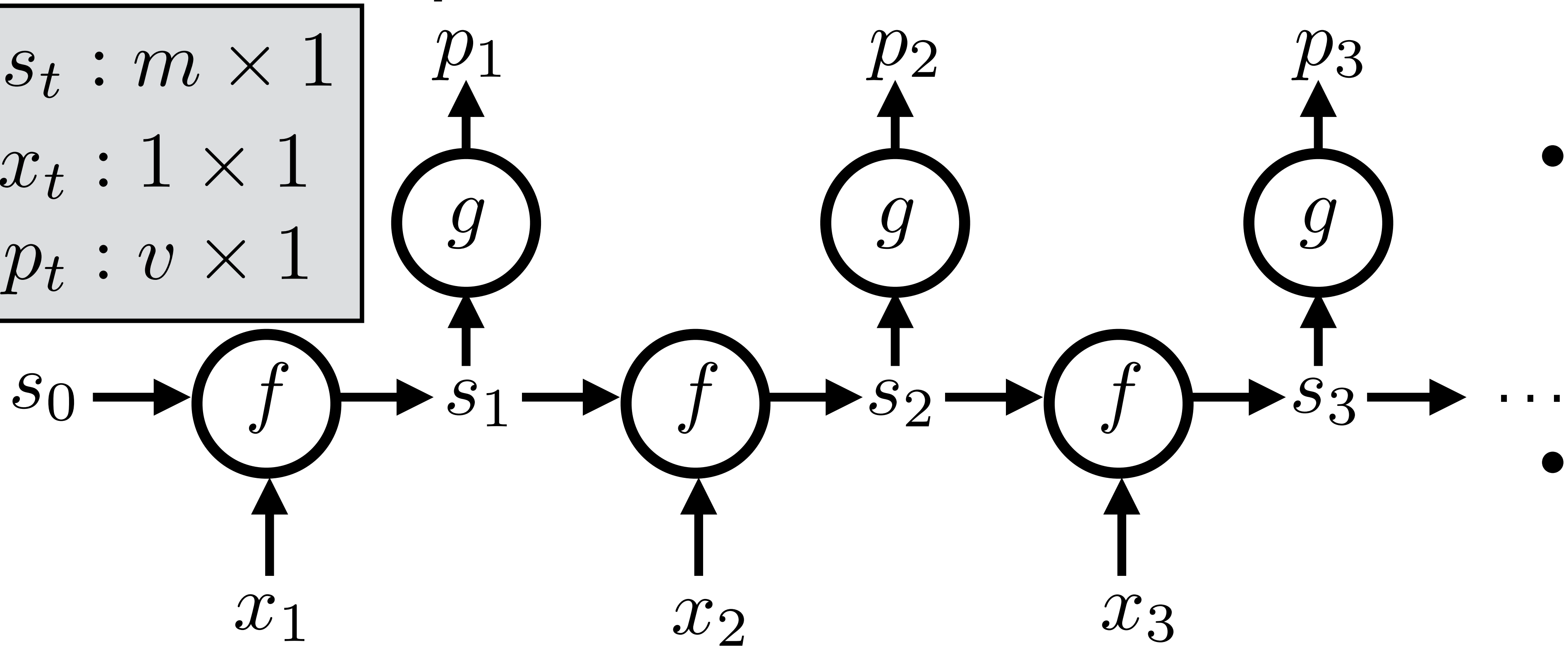
$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

v x *m* *v* x 1

v-class logistic regression

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1}; state is last *m* = 3 characters

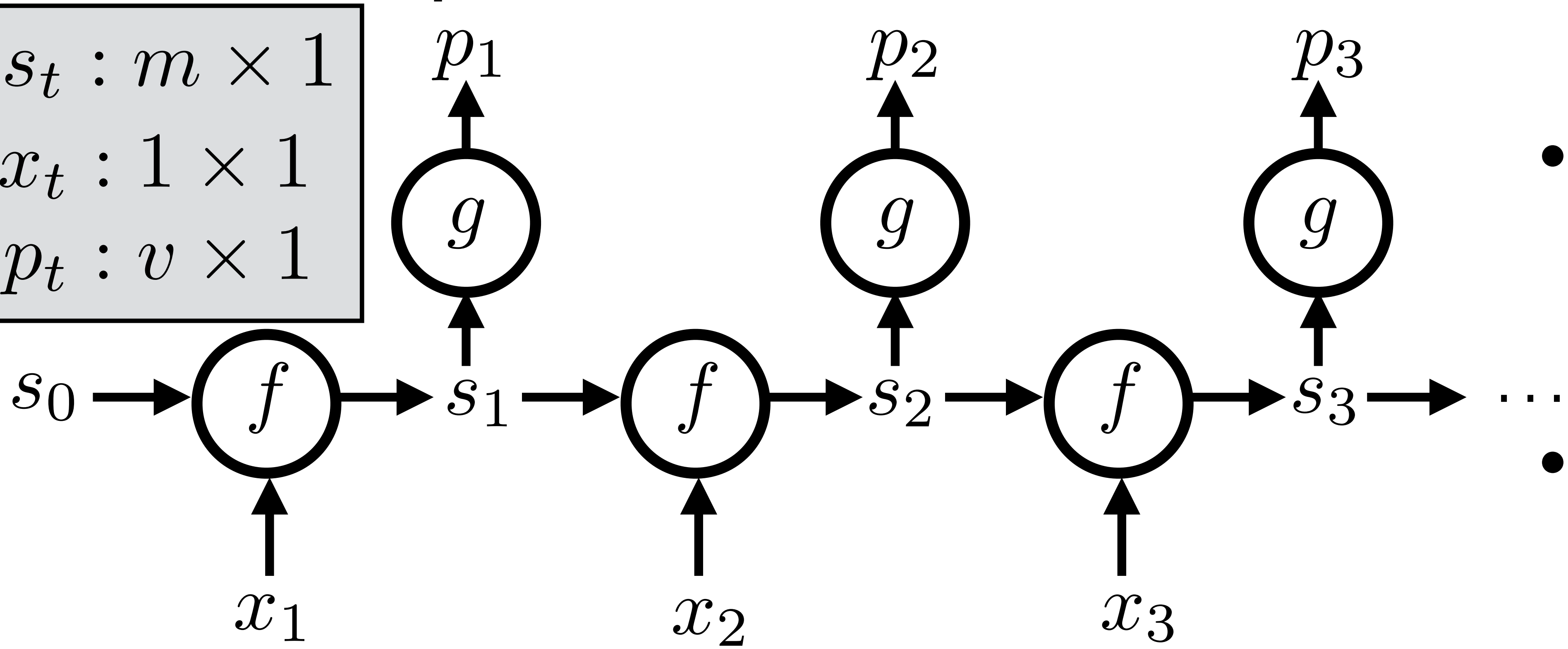
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

v-class logistic regression

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



- *m*: number of characters in the context
- *v*: number of characters in the alphabet

- Example: Alphabet {0,1};
  state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} \\ \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

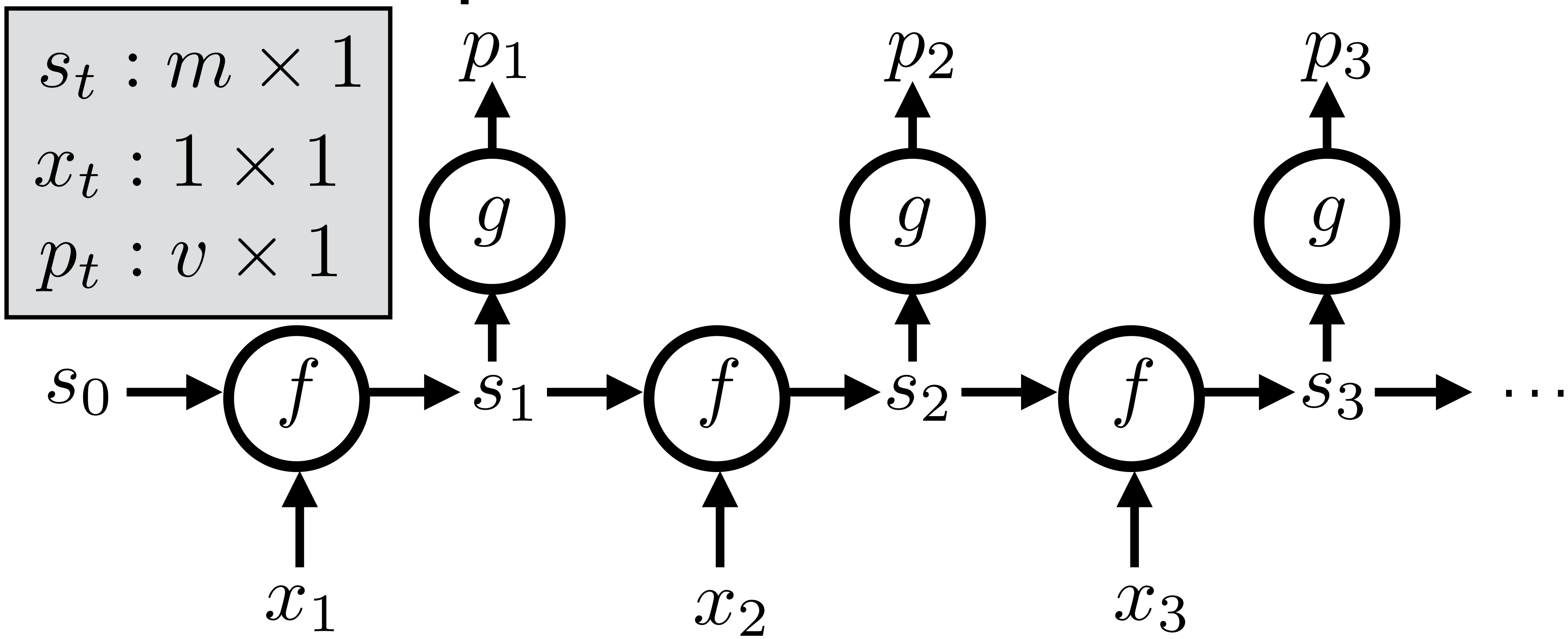
no transpose

$$p_t = g(s_t)$$
$$= f_2(\boxed{W^o} s_t + W_0^o)$$

$v \times m$ $\quad$ $v \times 1$

v-class logistic regression

Can express as a state machine



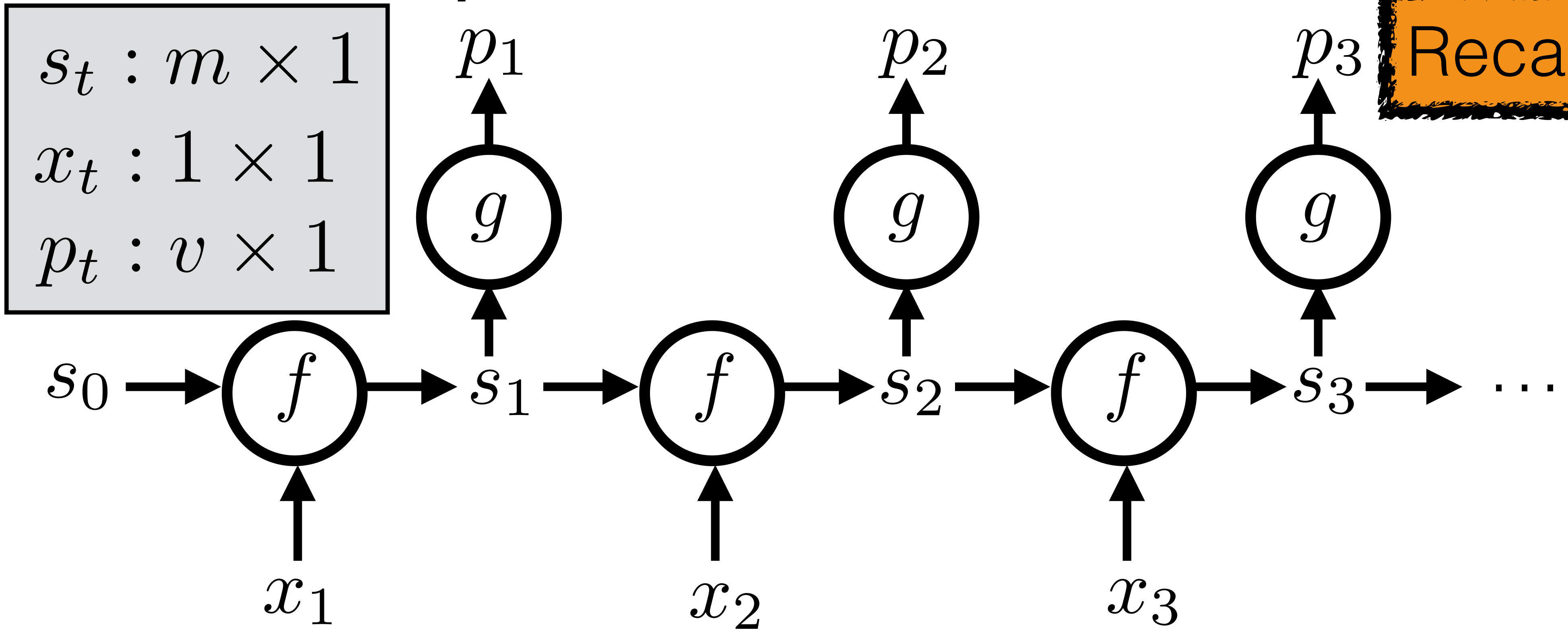
- Example: Alphabet {0,1};
  state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

v-class logistic regression

Can express as a state machine



Recall: familiar pattern

- Example: Alphabet {0,1}; state is last *m* = 3 characters

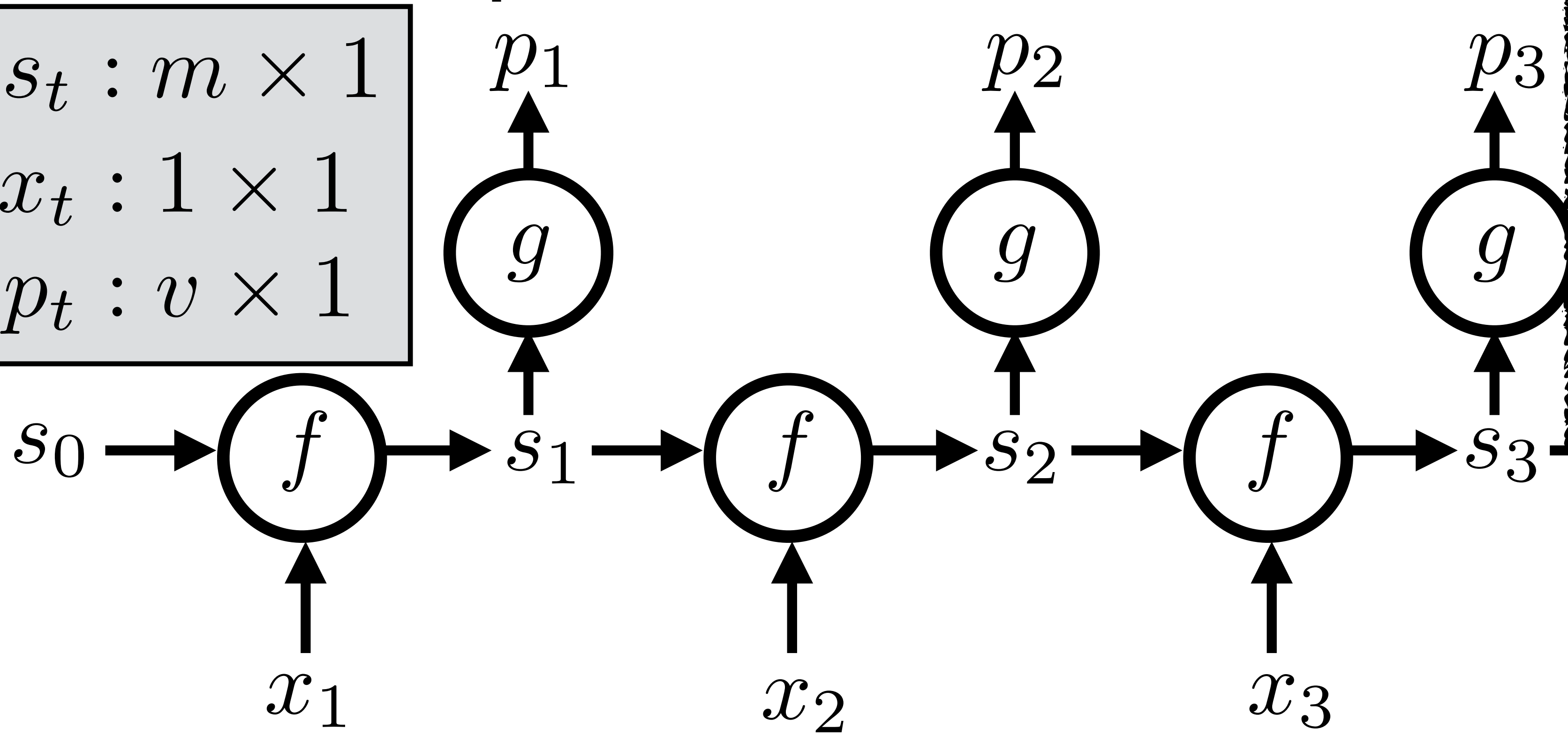
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underset{v \times m}{W^o} s_t + \underset{v \times 1}{W_0^o}) \end{aligned}$$

v-class logistic regression

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern
1. Choose how to predict label (given features & parameters)

- Example: Alphabet {0,1}; state is last m = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

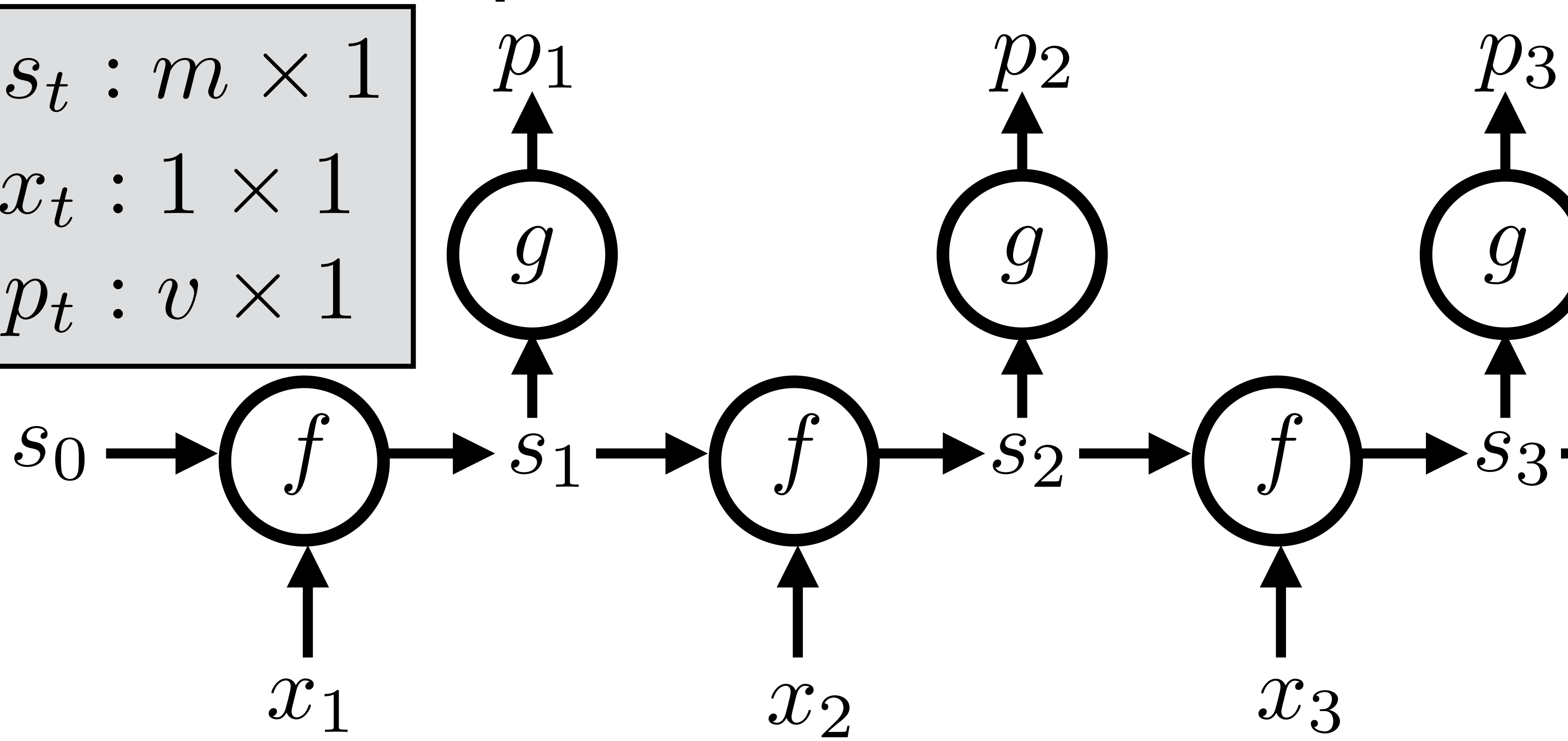
$$p_t = g(s_t)$$
$$= f_2(W^o s_t + W^o_0)$$

v x m $\quad$ v x 1

v-class logistic regression

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern
1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)

- Example: Alphabet {0,1}; state is last *m* = 3 characters

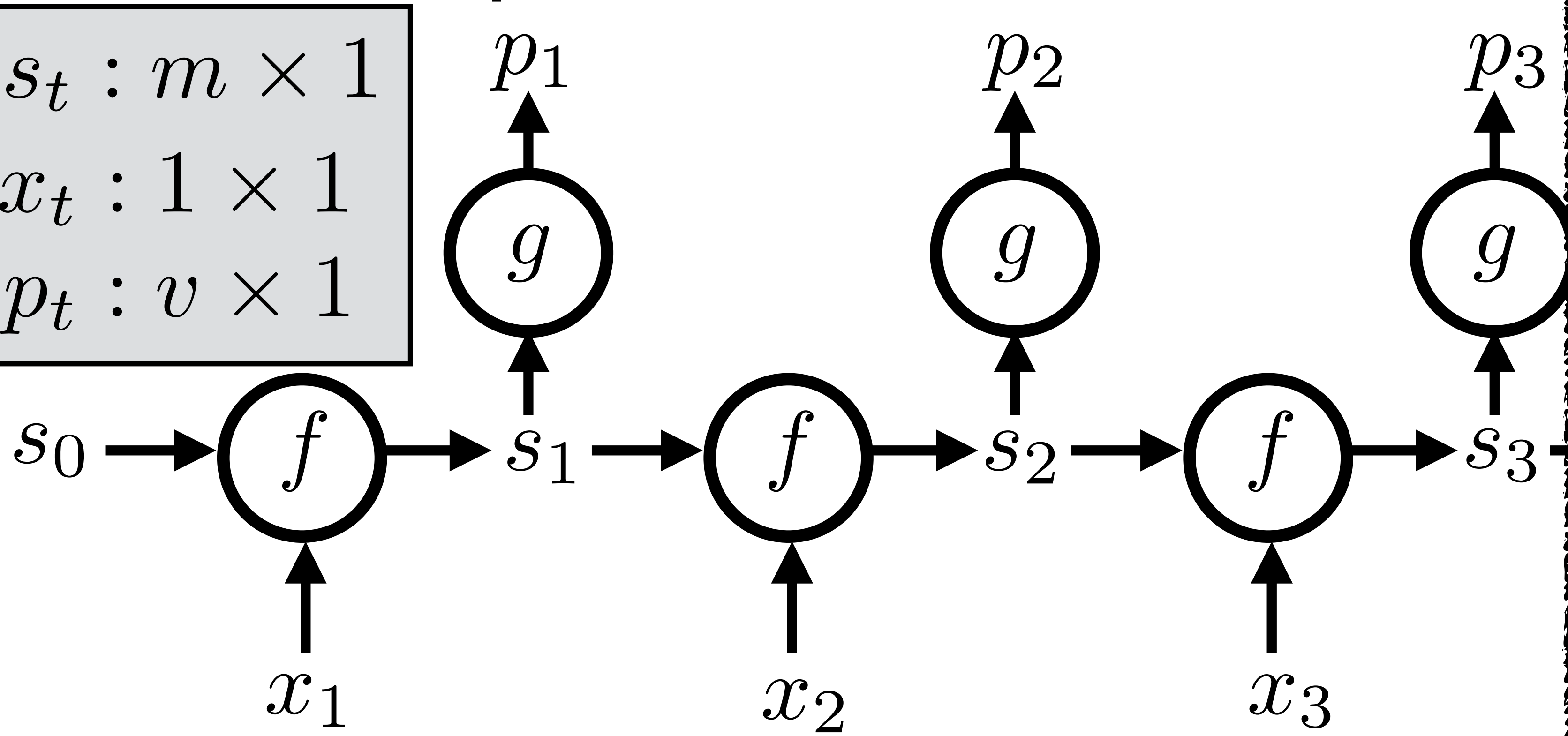
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W^o_0}_{v \times 1}) \end{aligned}$$

v-class logistic regression

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

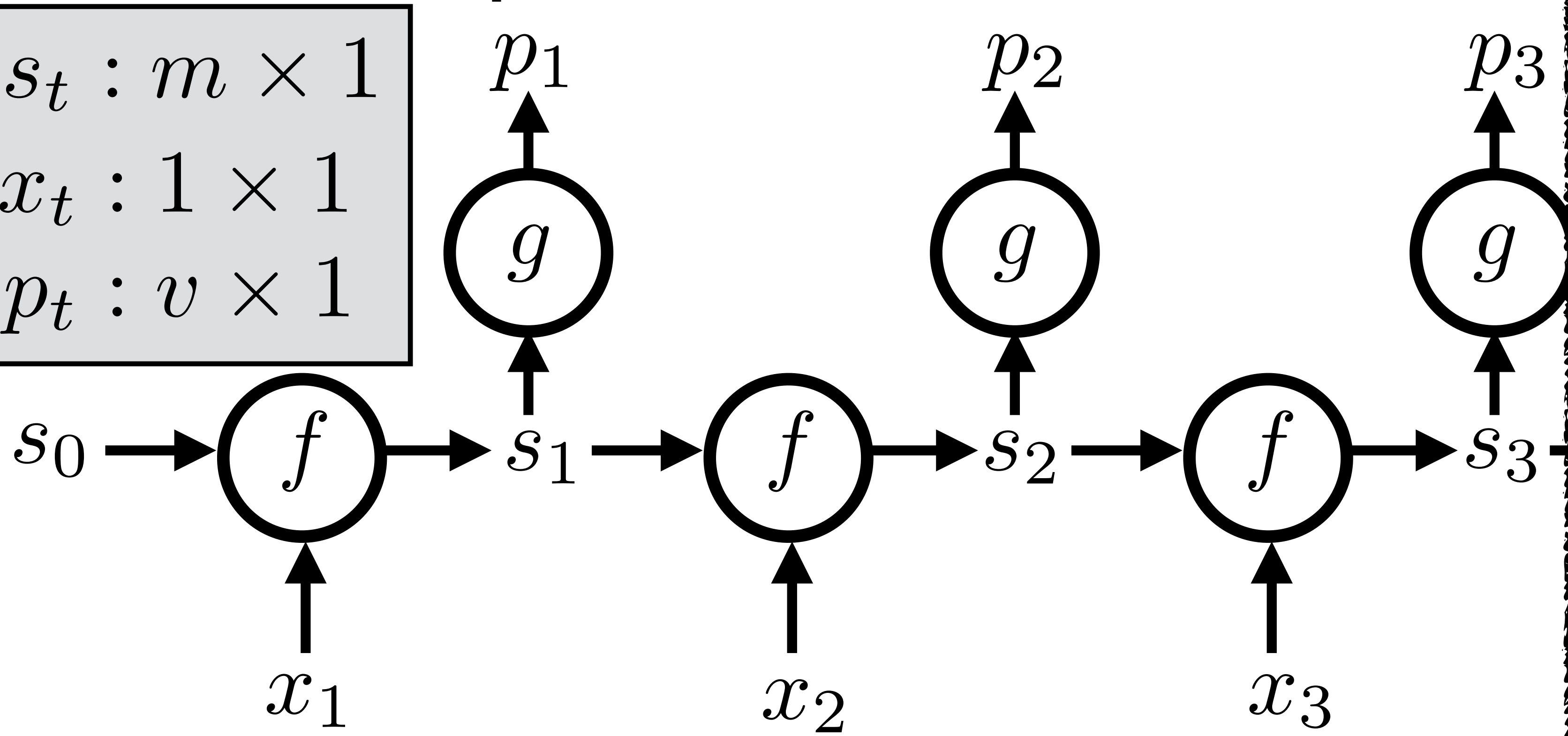
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

v-class logistic regression

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

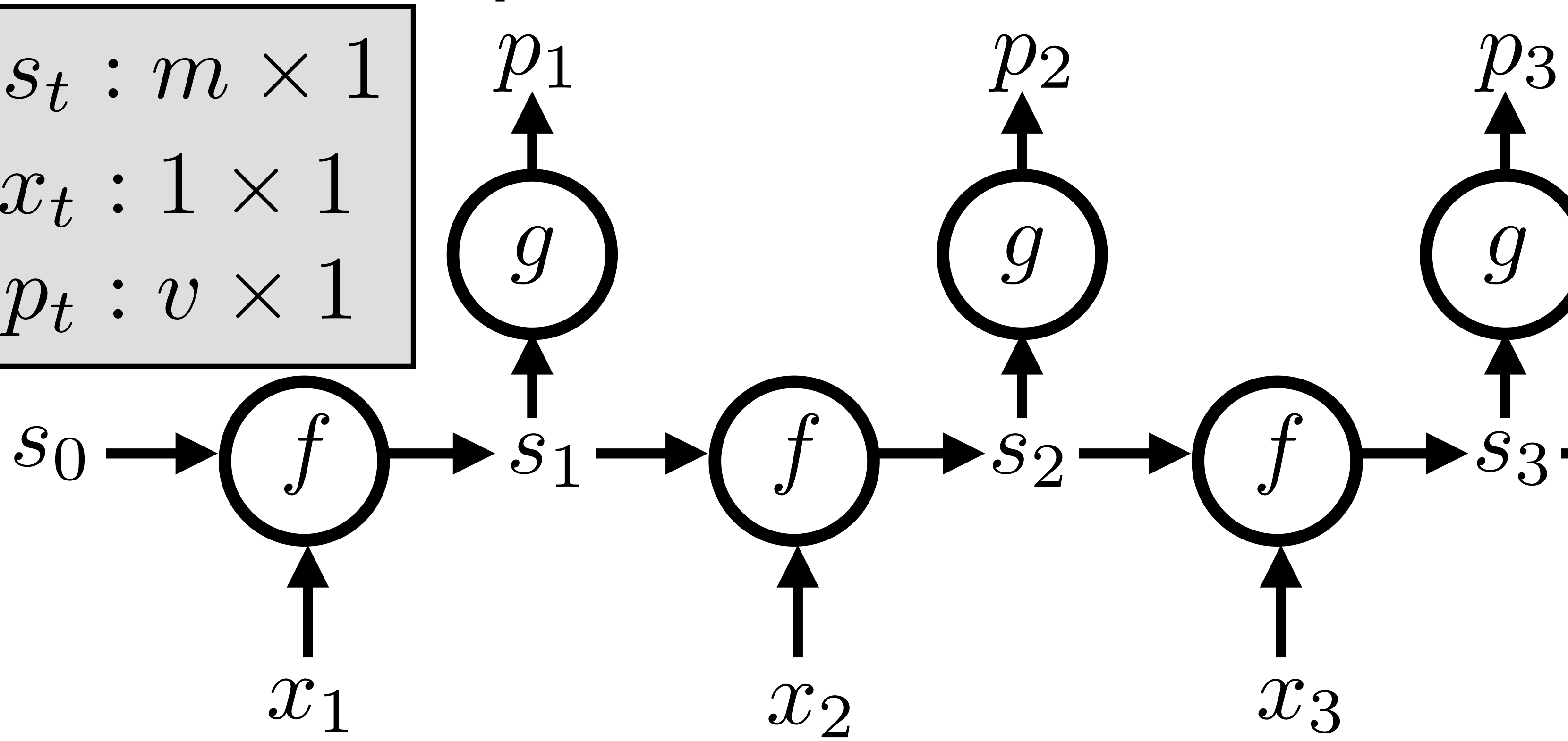
- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W^o_0}_{v \times 1}) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern
1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

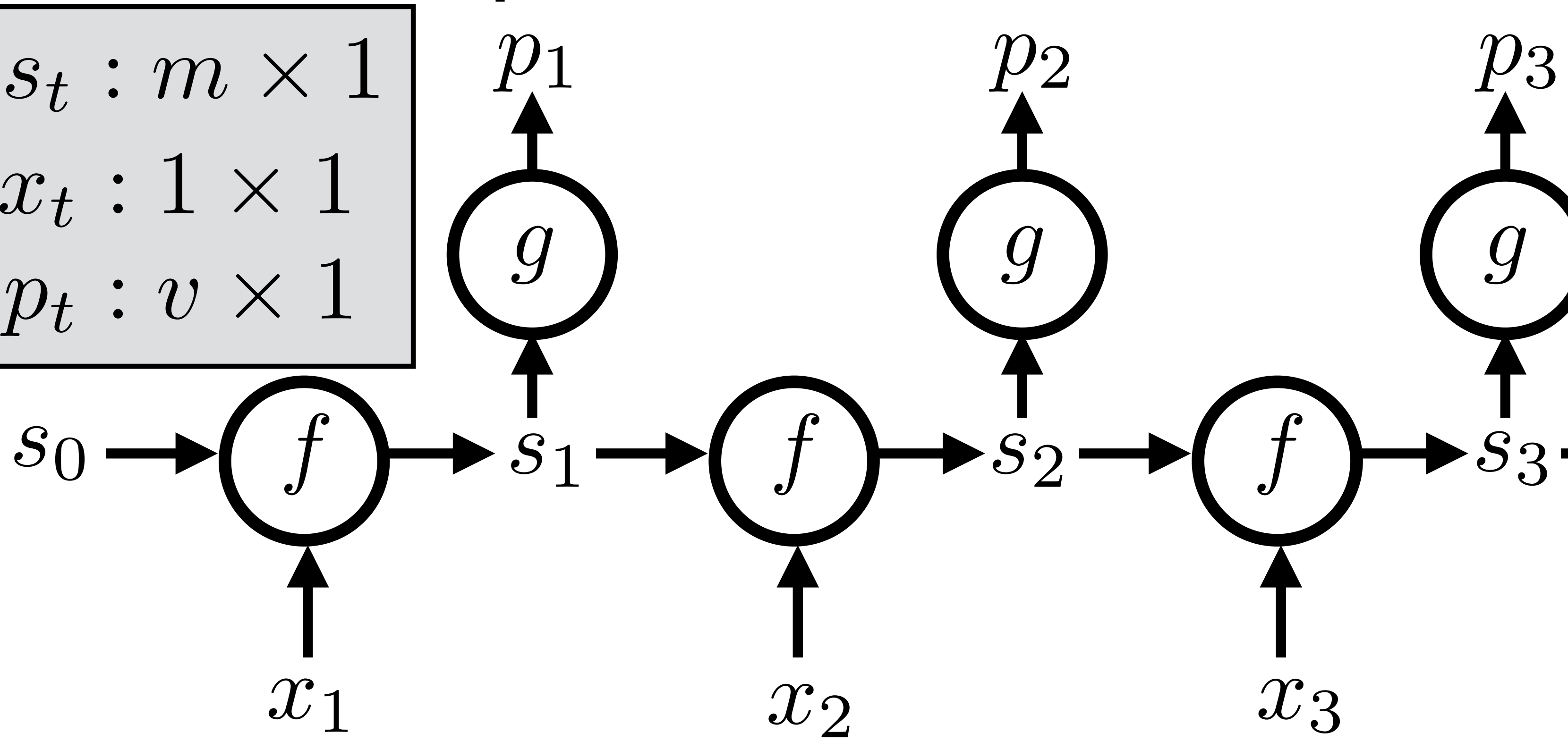
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$p_t = g(s_t) = f_2(W^o s_t + W_0^o)$$

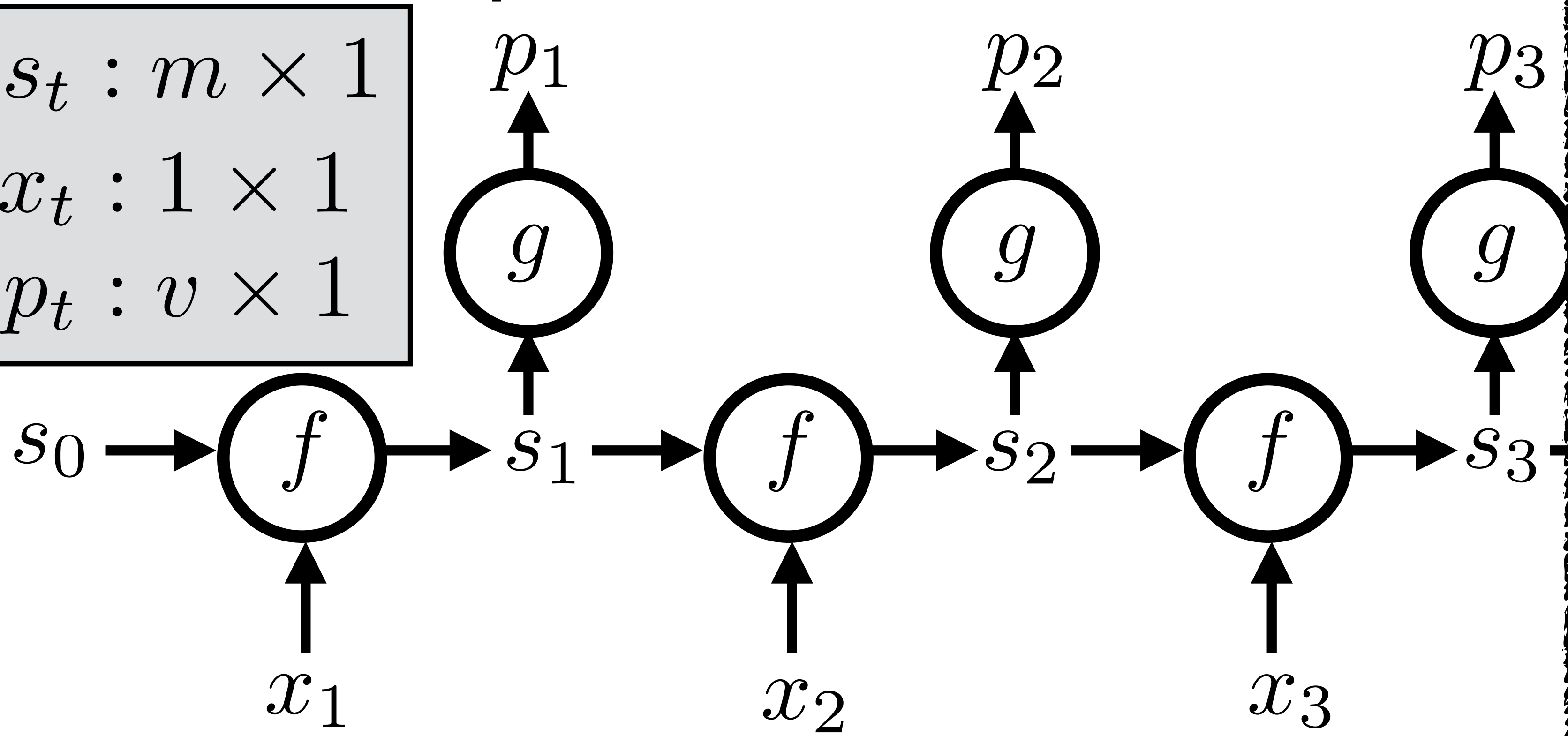
$v \times m$ $\quad$ $v \times 1$

$$L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$

$$p^{(i)} = \mathrm{R}(x^{(i)}; W^o, W_0^o)$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

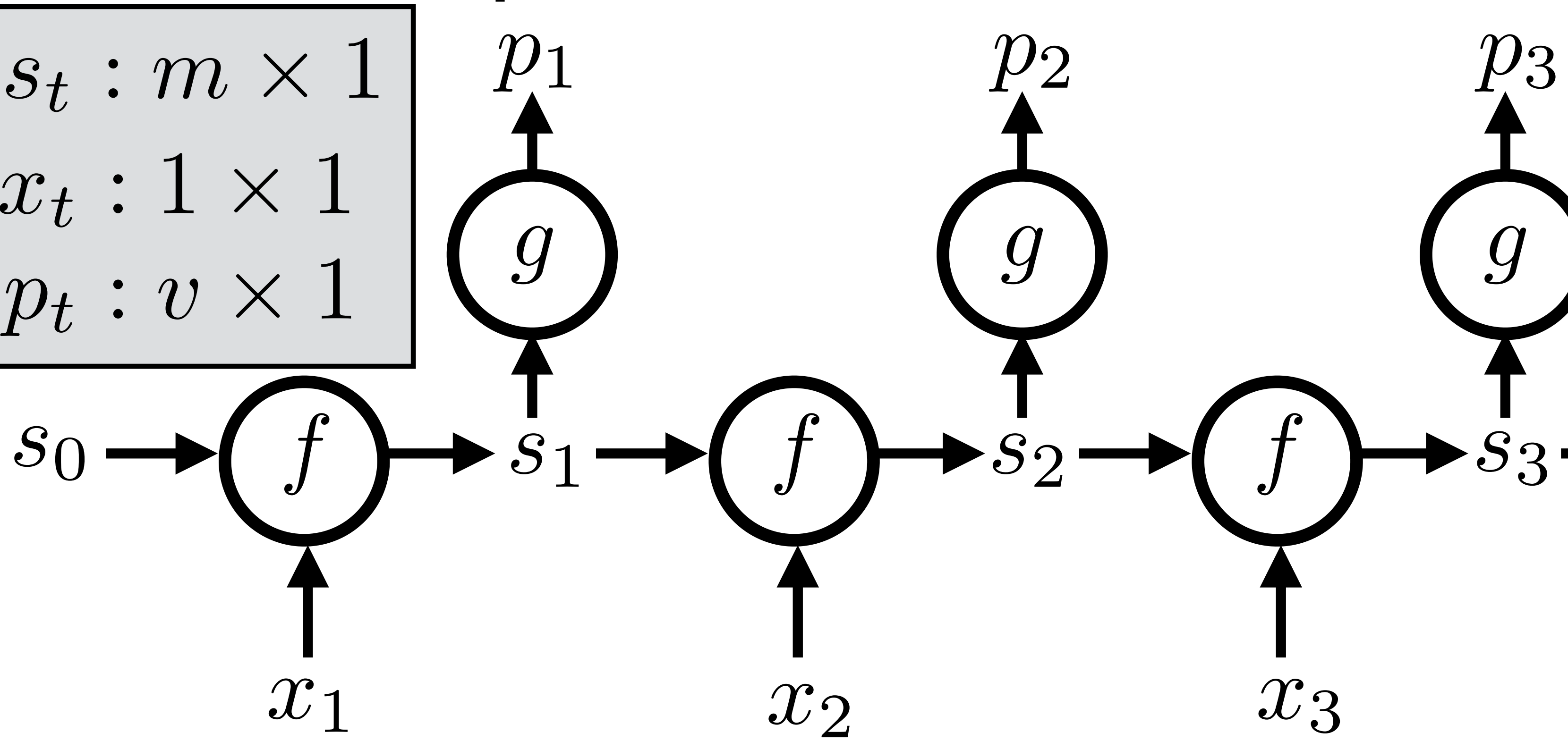
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underset{v \times m}{W^o} s_t + \underset{v \times 1}{W_0^o}) \end{aligned}$$

$$L_{\text{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$
$$p^{(i)} = \mathrm{R}(x^{(i)}; W^o, W_0^o)$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

$v \times m$ $\quad$ $v \times 1$

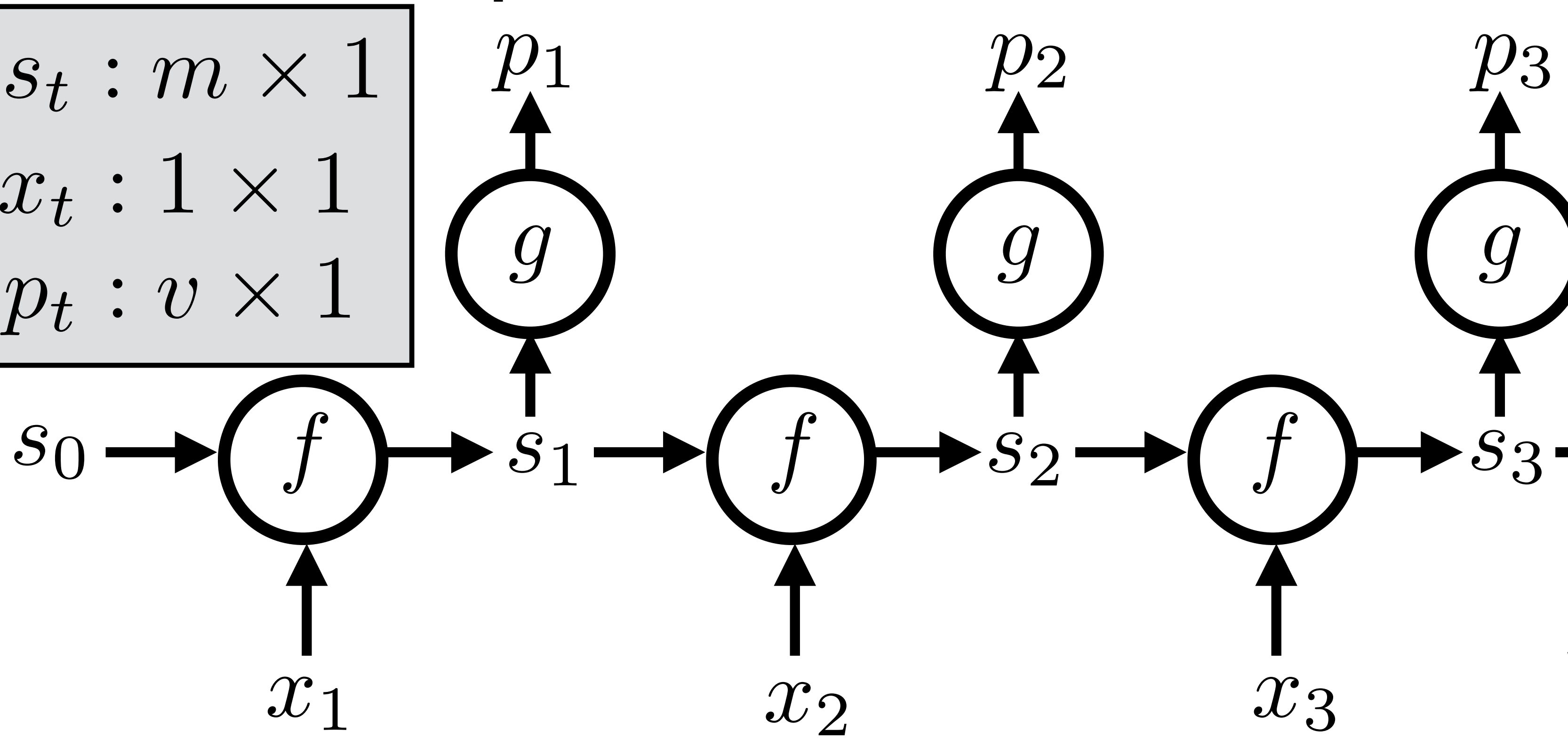
$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$

$x_t : 1 \times 1$

$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

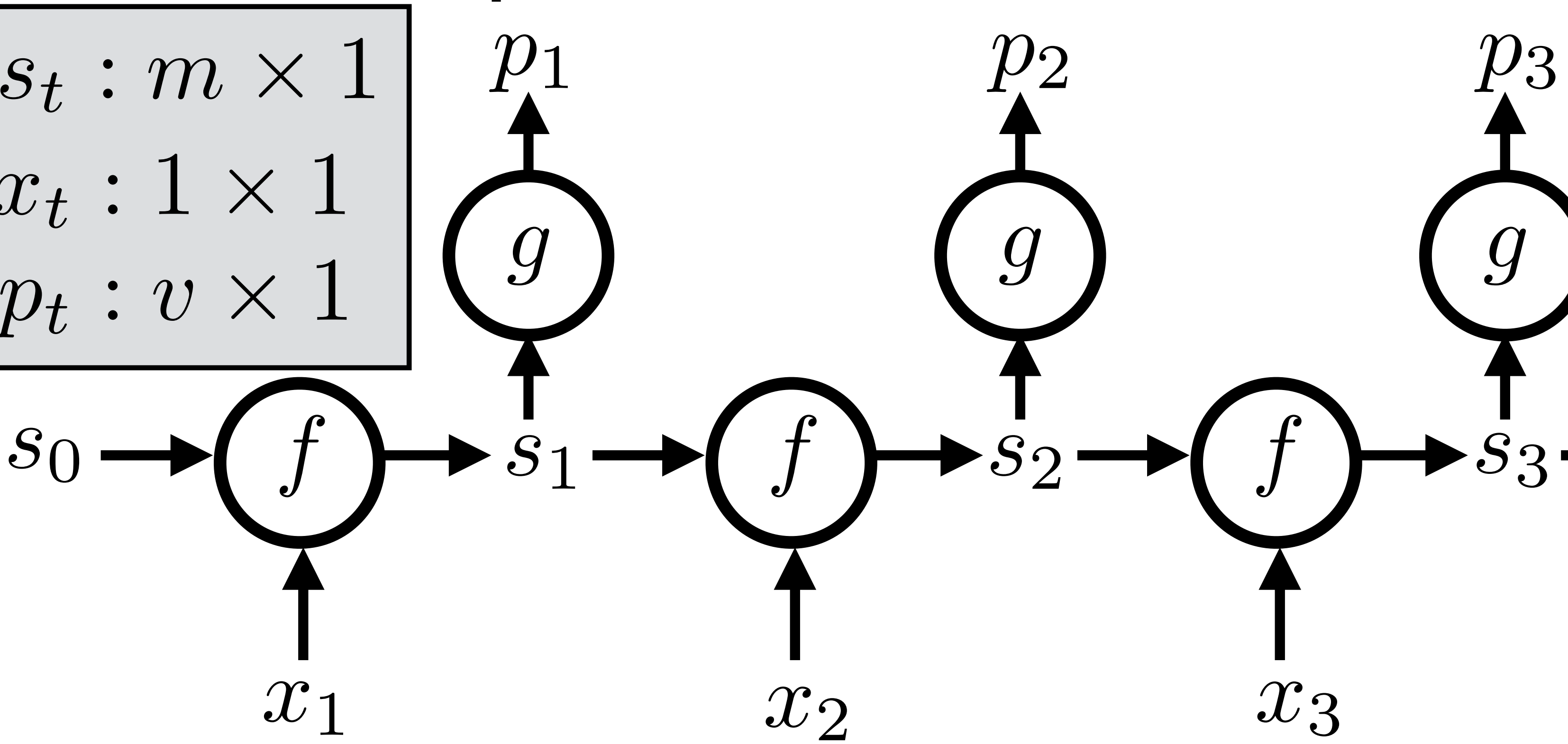
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last m = 3 characters

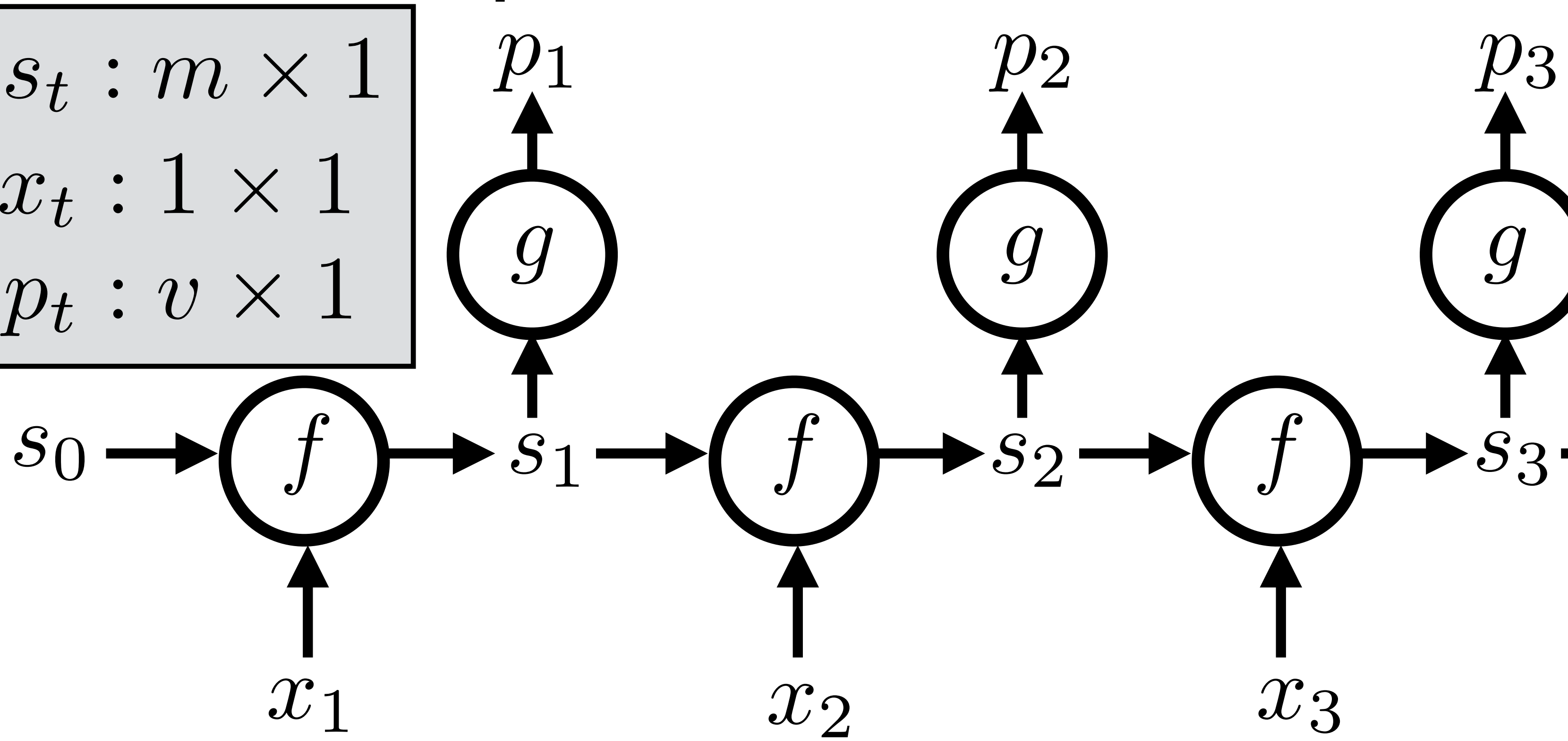
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W^o_0}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W^o_0) \\ J(W^o, W^o_0) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

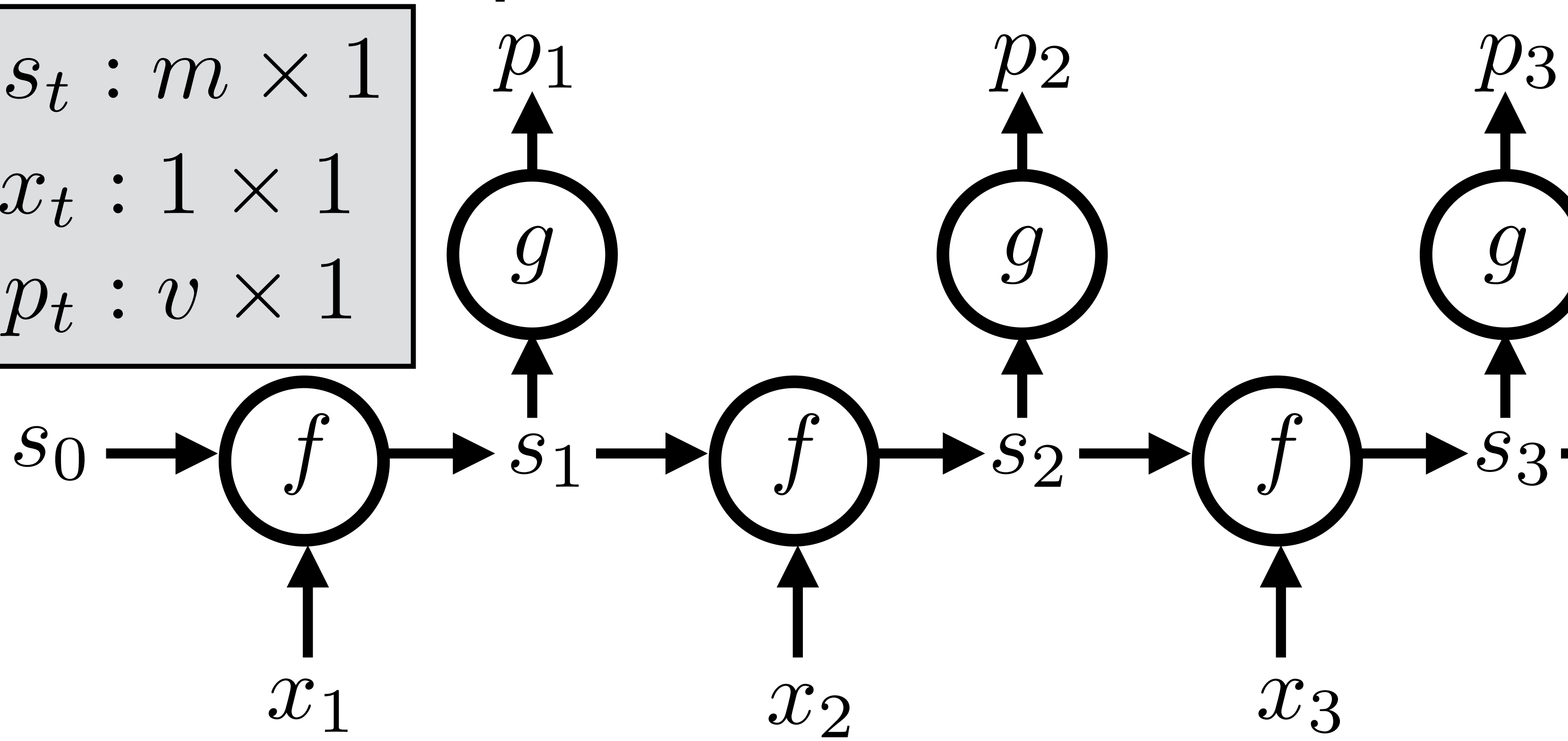
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W^o_0}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W^o_0) \\ J(W^o, W^o_0) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$$s_t : m \times 1$$
$$x_t : 1 \times 1$$
$$p_t : v \times 1$$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

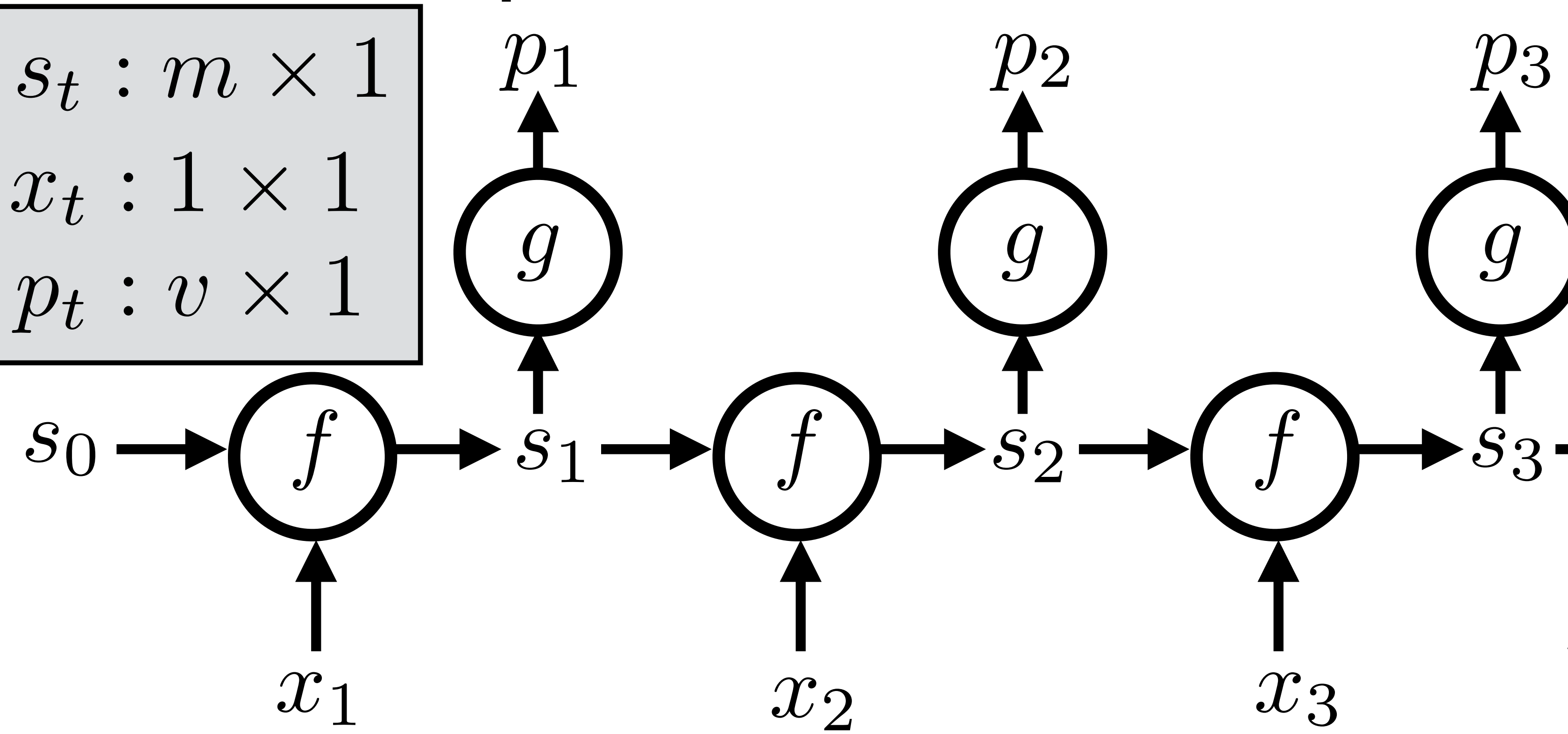
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1}$$

$$p_t = g(s_t) = f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1})$$

$$L_{\text{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$
$$p^{(i)} = \mathrm{R}(x^{(i)}; W^o, W_0^o)$$
$$J(W^o, W_0^o) = \sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right)$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern
1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1}$$

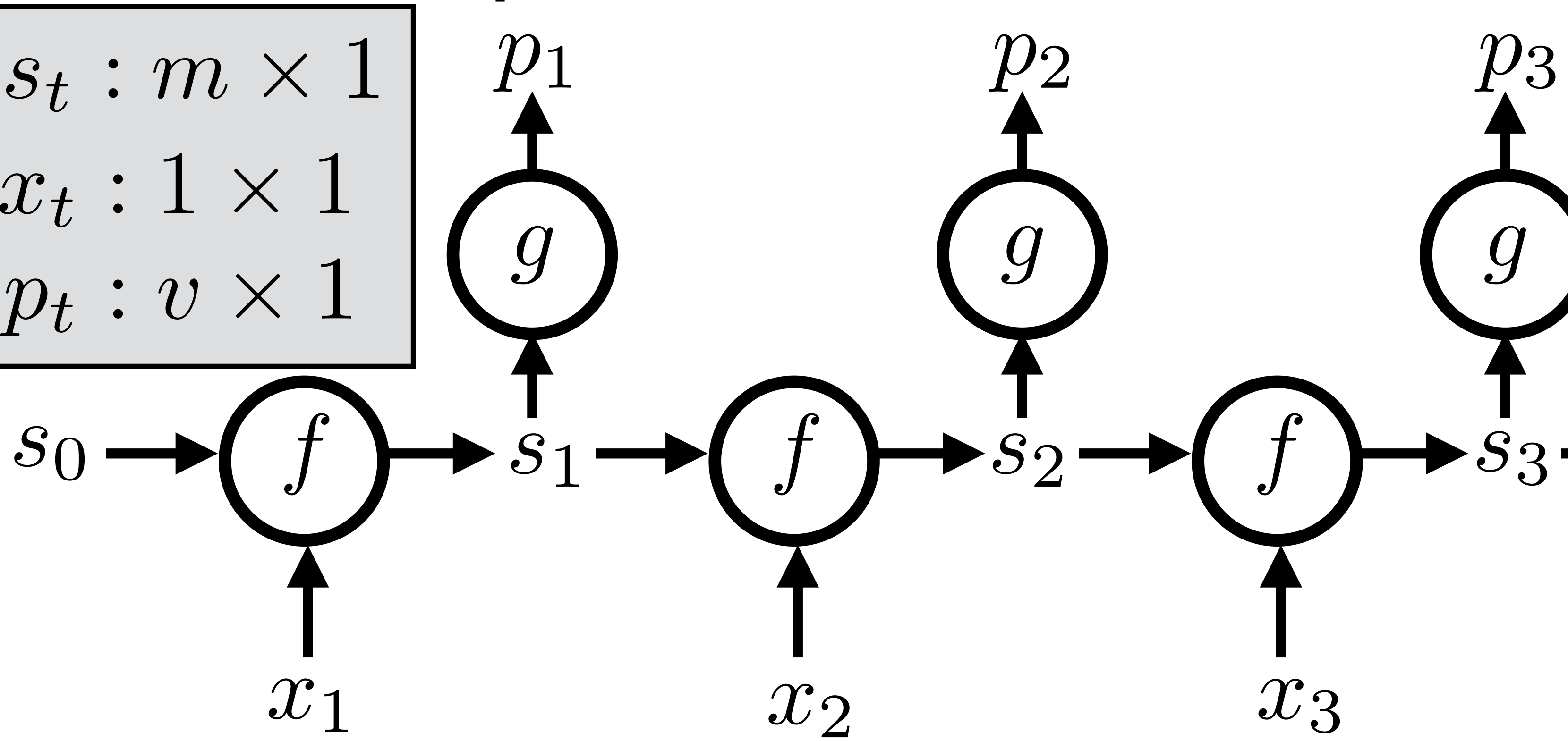
$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

$v \times m$ $\quad$ $v \times 1$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

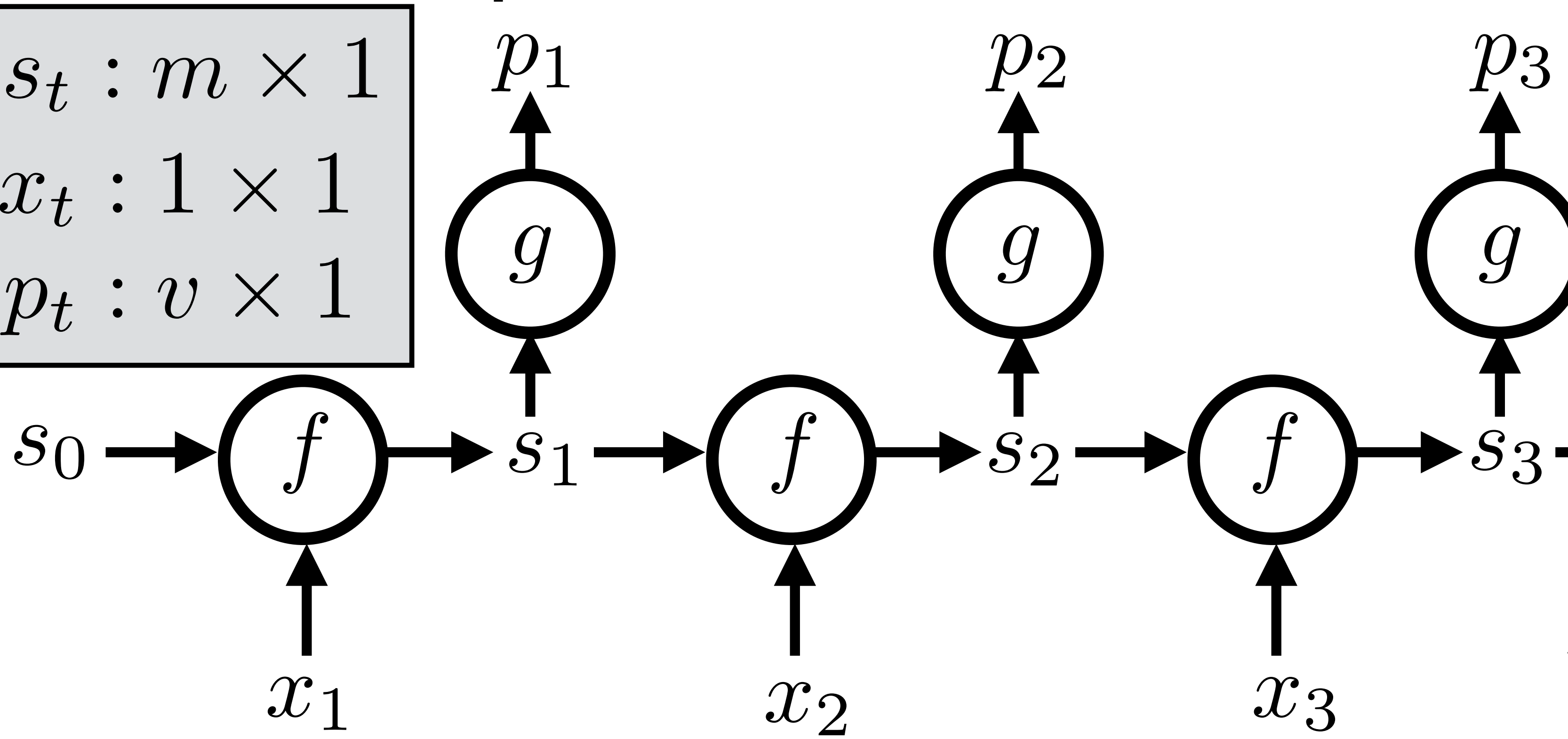
$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W^o_0}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W^o_0) \\ J(W^o, W^o_0) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$$s_t : m \times 1$$
$$x_t : 1 \times 1$$
$$p_t : v \times 1$$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

v x *m*   *v* x 1

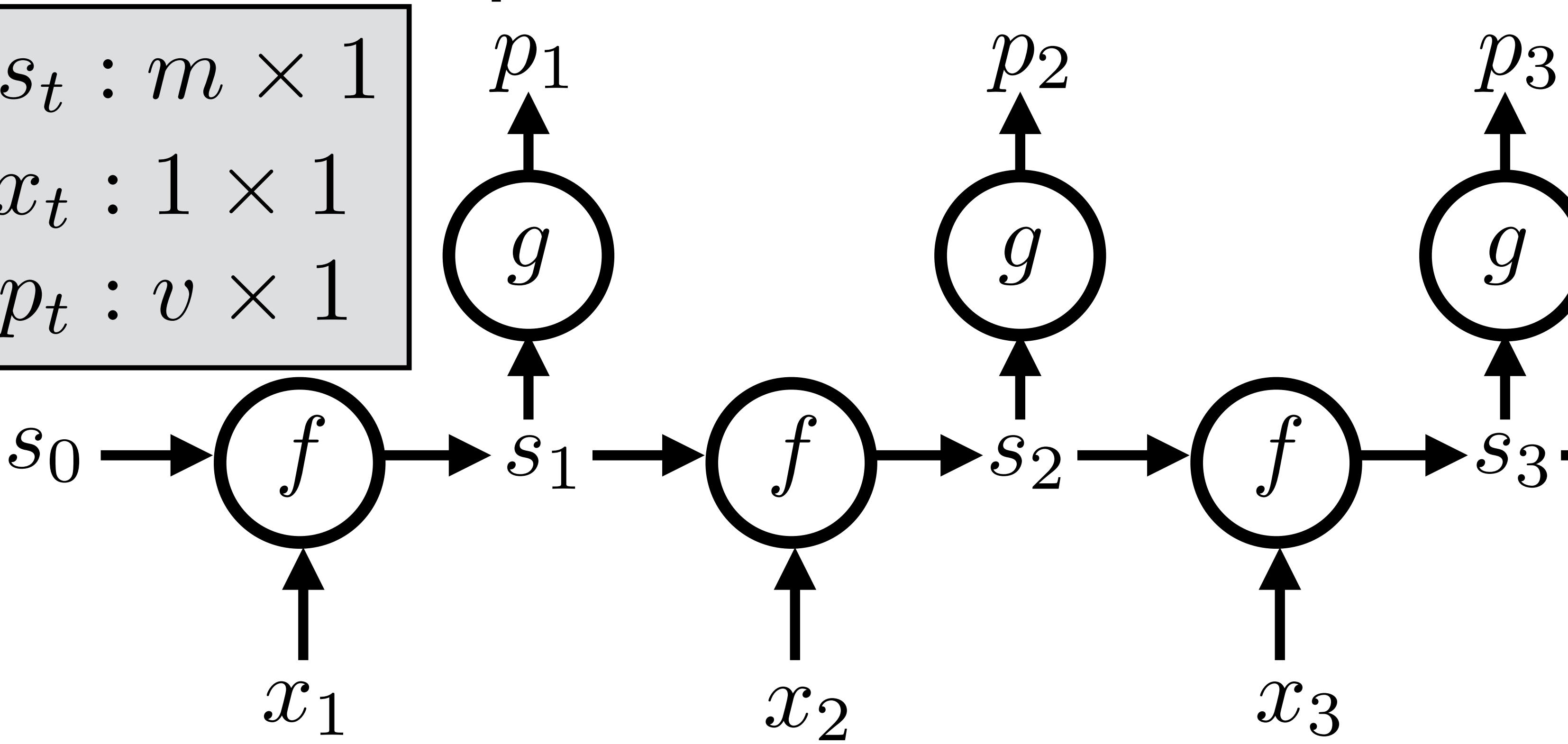
$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$

$x_t : 1 \times 1$

$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

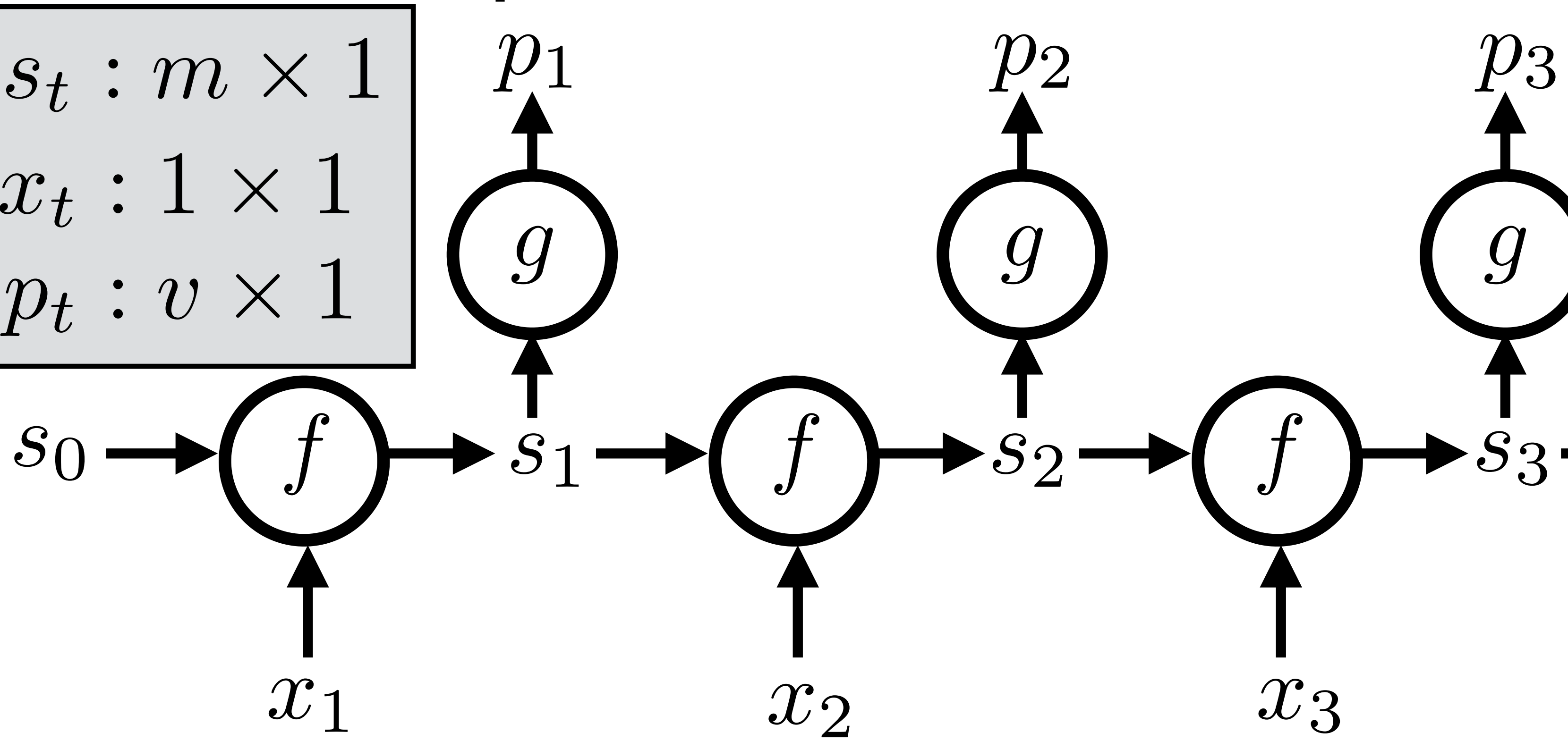
$$s_t = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

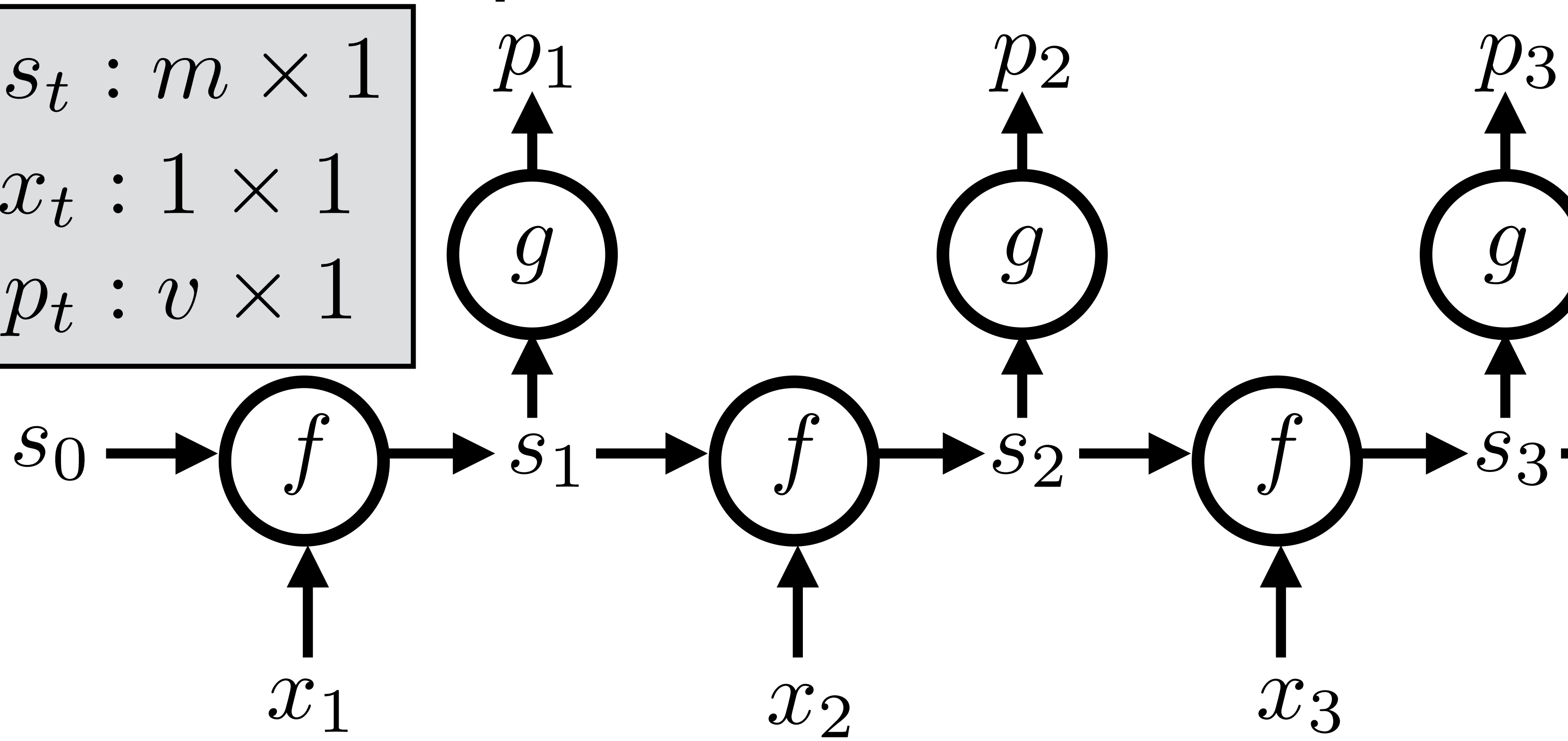
$$s_t = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1} + \begin{bmatrix} w_{0,1}^{ss} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern
1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

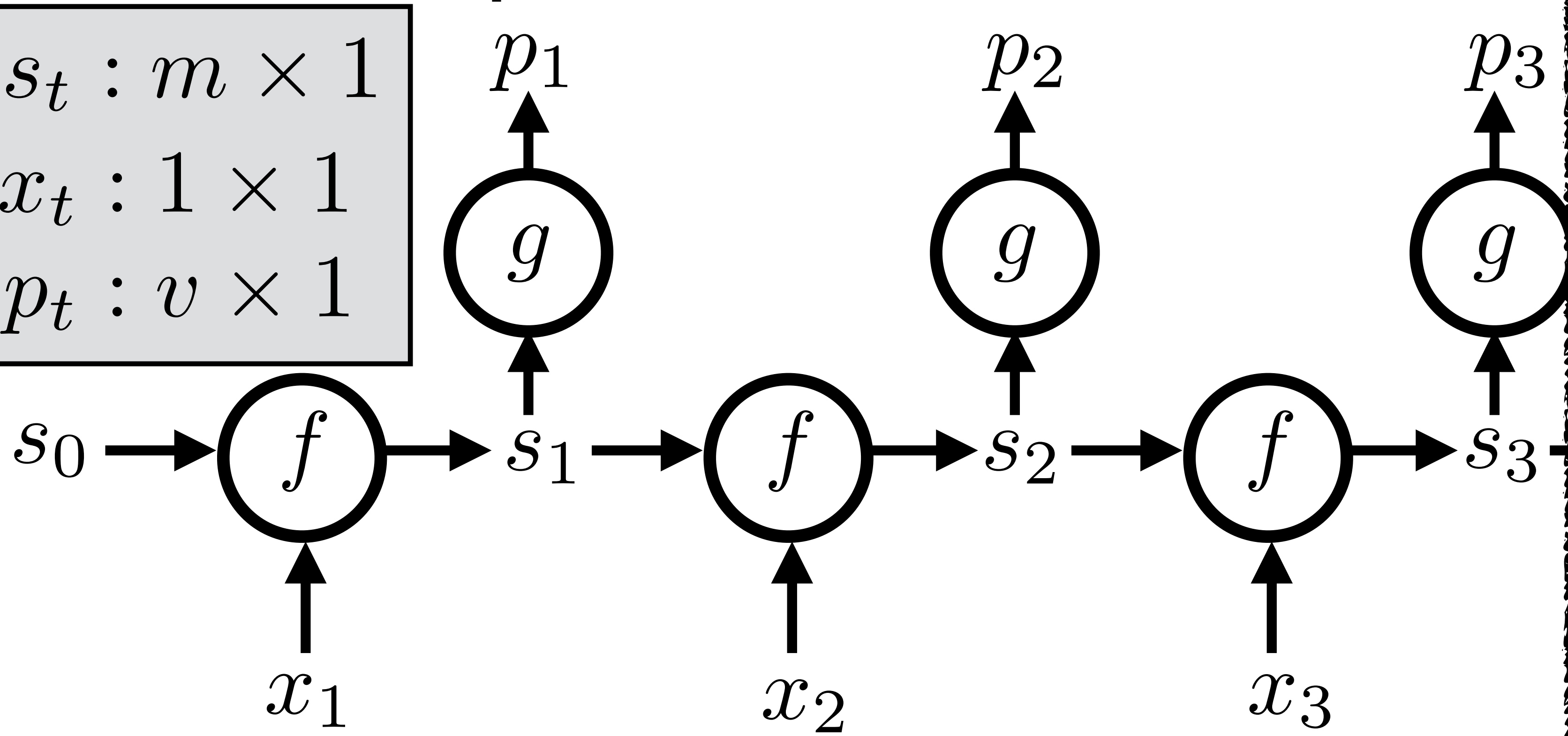
$$s_t = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1} + \begin{bmatrix} w_{0,1}^{ss} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

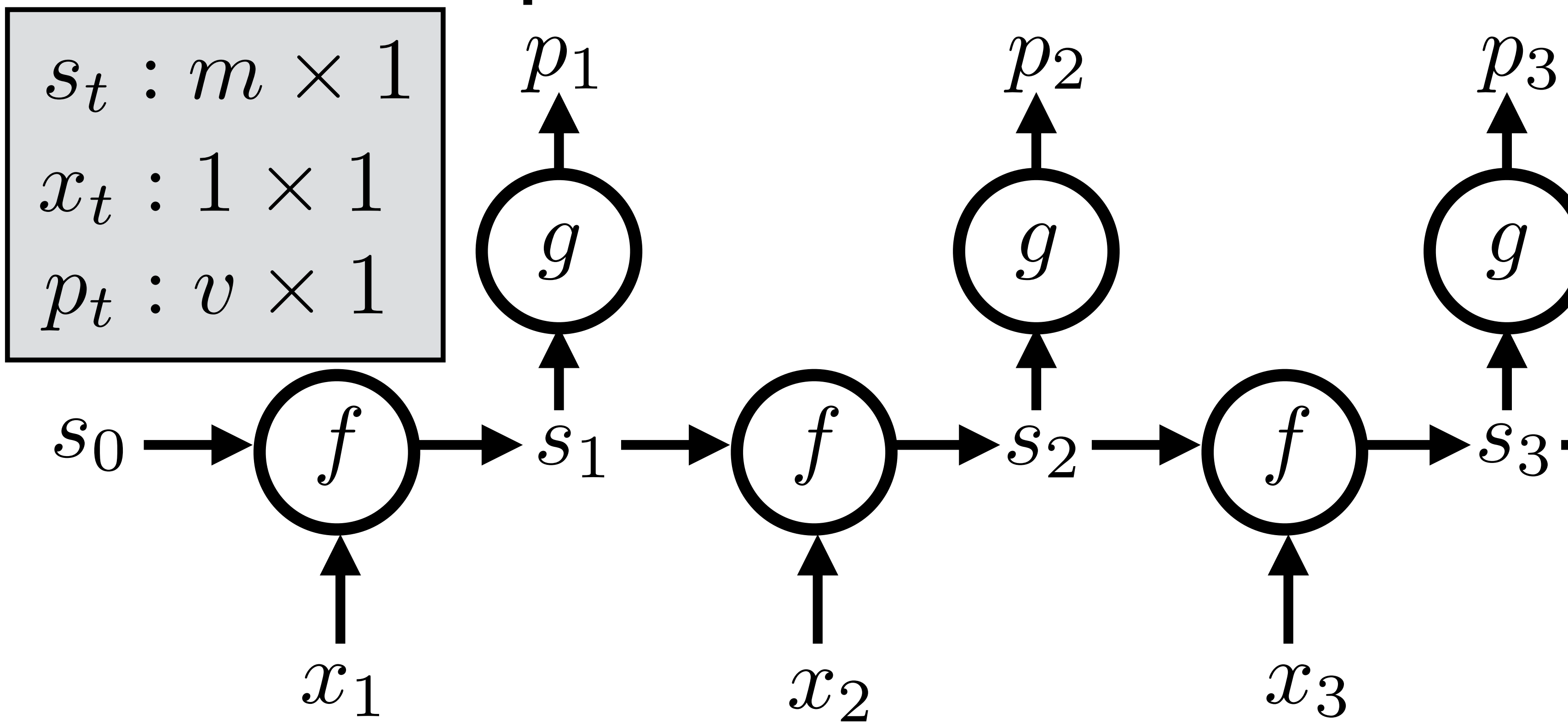
- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} \boxed{x_t} + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} \boxed{s_{t-1}} + \begin{bmatrix} w_{0,1}^{ss} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

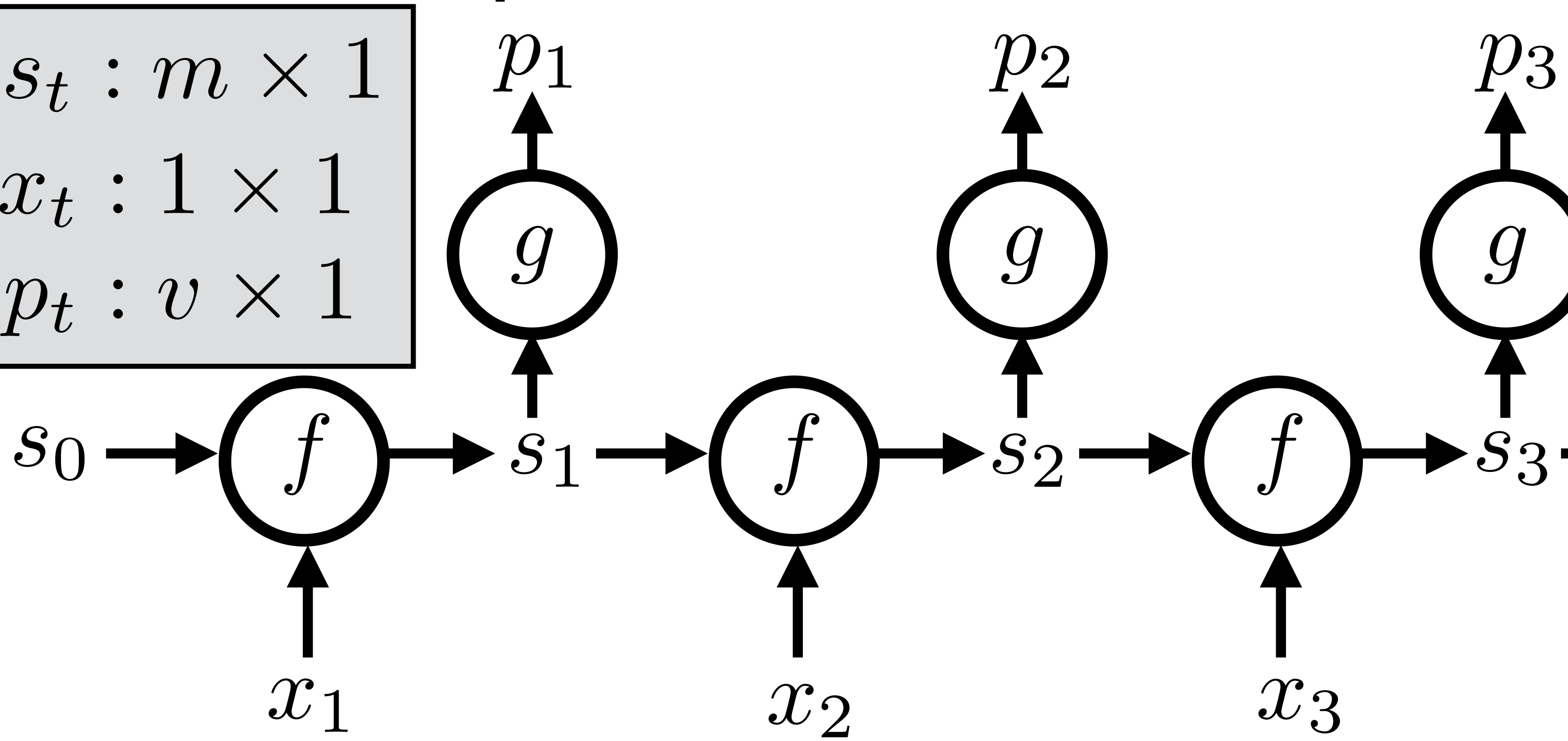
$$s_t = \begin{bmatrix} \boxed{w_1^{sx}} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} \boxed{x_t} + \begin{bmatrix} \boxed{w_{11}^{ss} \quad w_{12}^{ss} \quad w_{13}^{ss}} \\ w_{21}^{ss} \quad w_{22}^{ss} \quad w_{23}^{ss} \\ w_{31}^{ss} \quad w_{32}^{ss} \quad w_{33}^{ss} \end{bmatrix} \boxed{s_{t-1}} + \begin{bmatrix} \boxed{w_{0,1}^{ss}} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

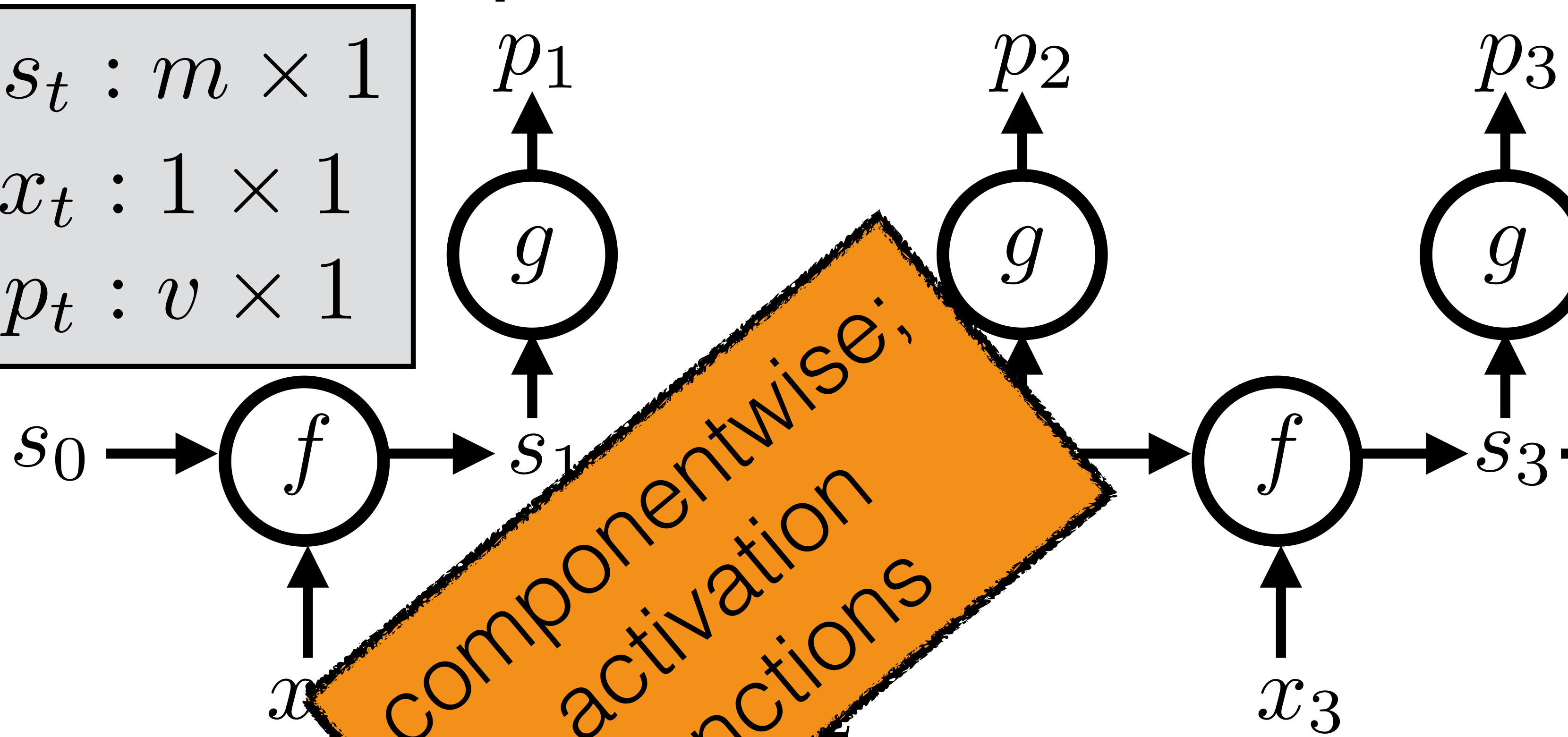
$$s_t = f_1\left(\begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1} + \begin{bmatrix} w_{0,1}^{ss} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

$$L_{\text{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$
$$p^{(i)} = \mathrm{R}(x^{(i)}; W^o, W_0^o)$$
$$J(W^o, W_0^o) = \sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right)$$

Can express as a state machine

$$s_t : m \times 1$$
$$x_t : 1 \times 1$$
$$p_t : v \times 1$$



componentwise; activation functions

Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Exampl… …et {0,1}; state is la… = 3 characters

$$s_t = f_1\left(\begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1} + \begin{bmatrix} w_{0,1}^{ss} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix} \right)$$

$$p_t = g(s_t)$$
$$= f_2(W^o s_t + W_0^o)$$

v x m   v x 1

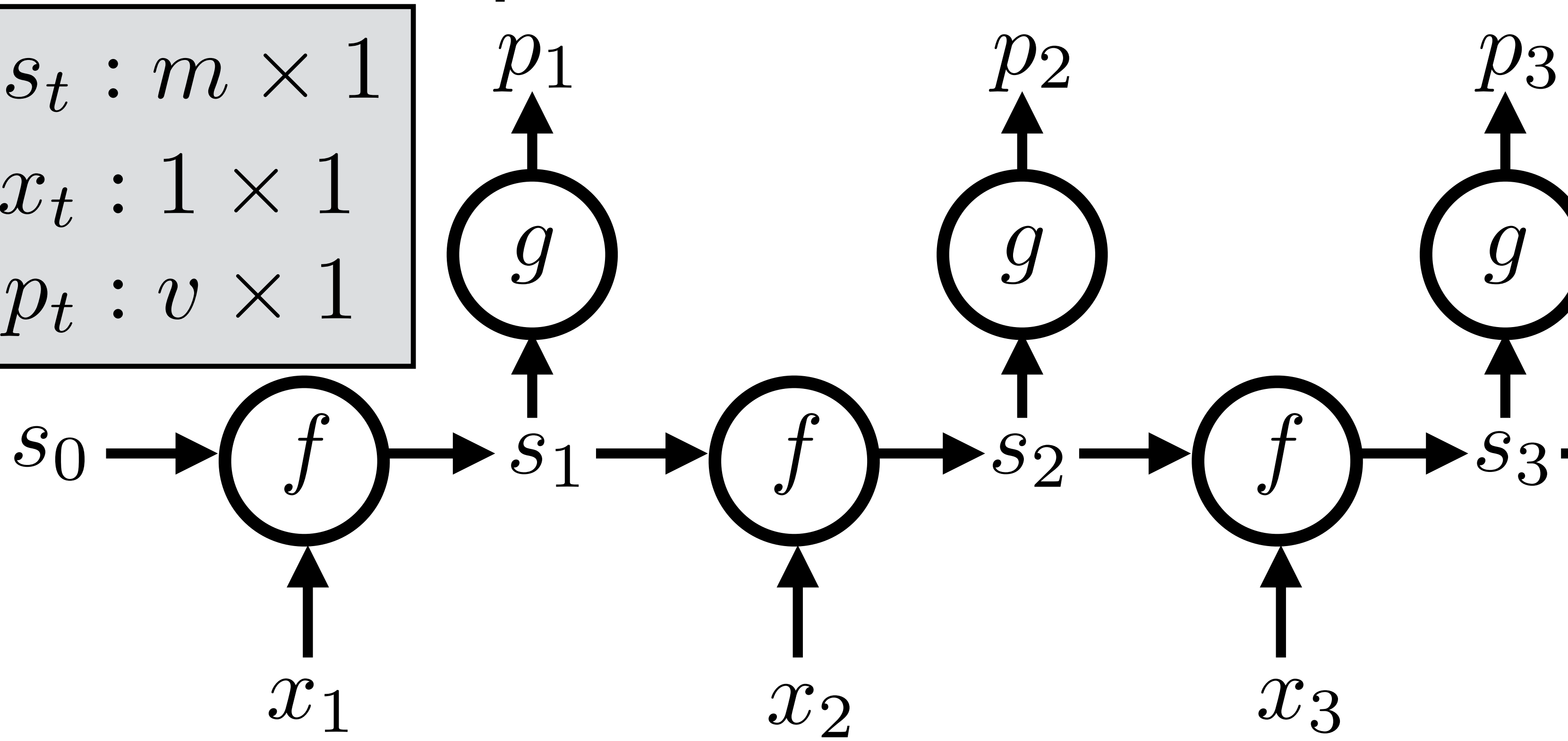
$$L_{\text{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$
$$p^{(i)} = \mathrm{R}(x^{(i)}; W^o, W_0^o)$$
$$J(W^o, W_0^o) = \sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right)$$

Can express as a state machine

$s_t : m \times 1$

$x_t : 1 \times 1$

$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

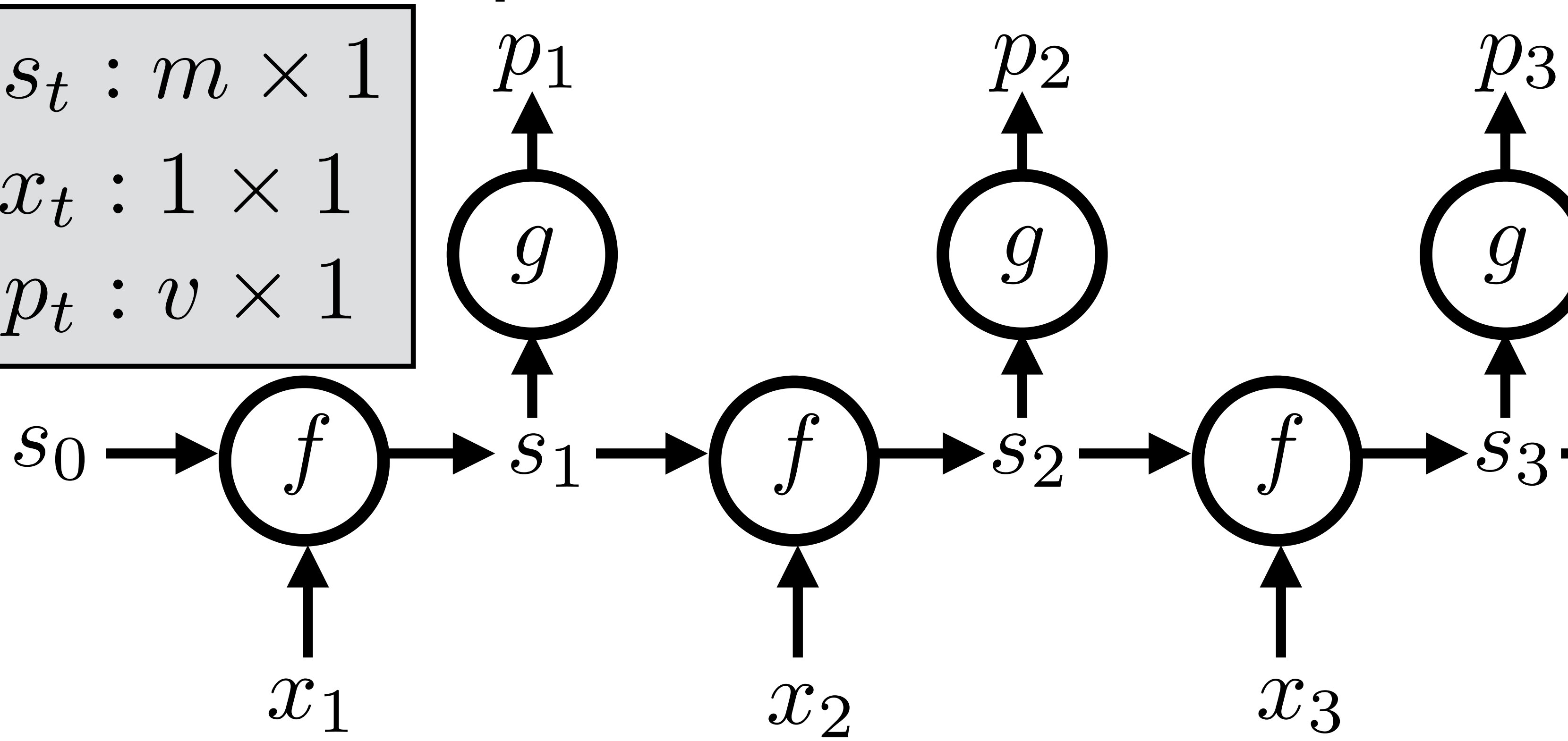
$$s_t = f_1\left(\begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1} + \begin{bmatrix} w_{0,1}^{ss} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern
1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

$$s_t = f_1\left(\begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1} + \begin{bmatrix} w_{0,1}^{ss} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix}\right)$$

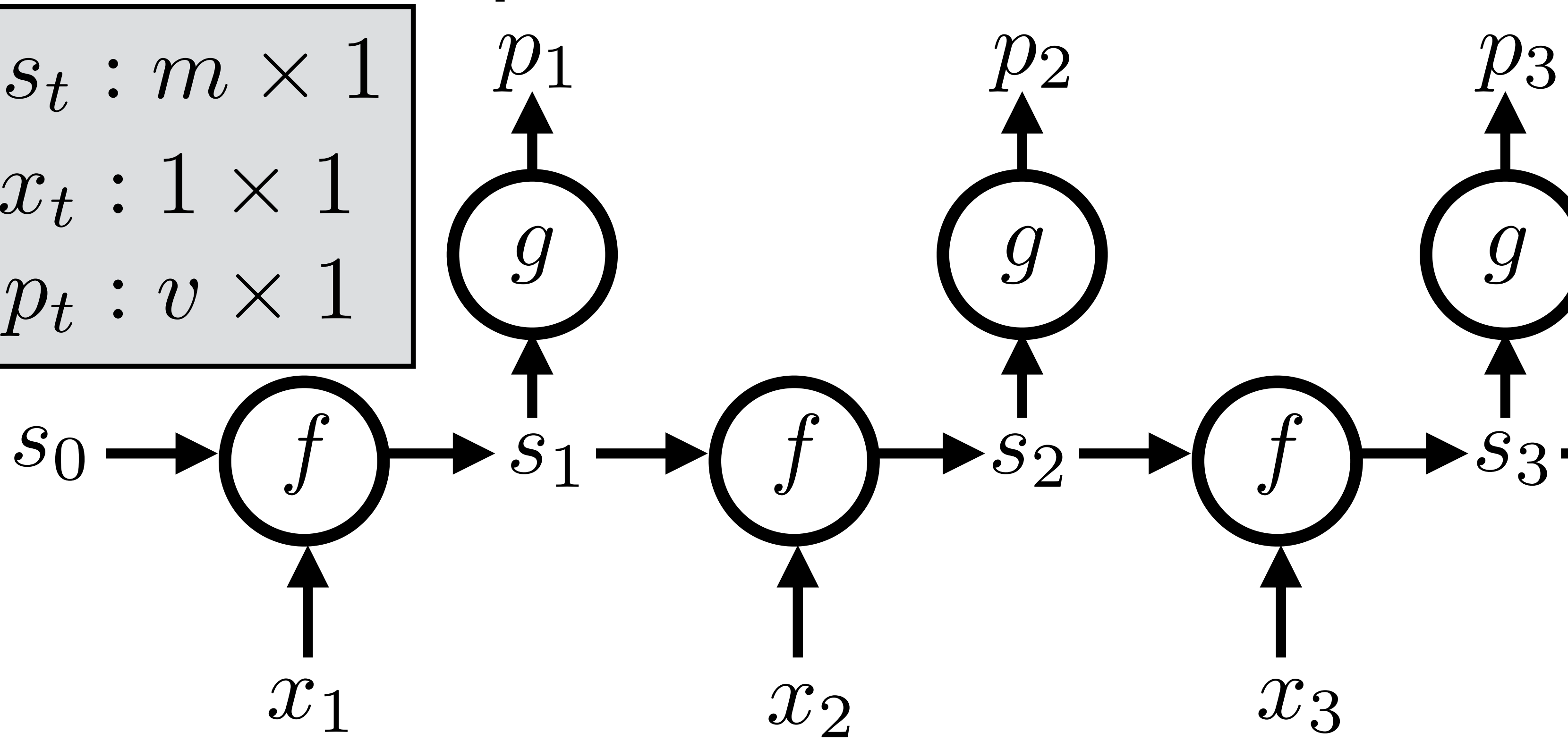
$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

$v \times m$ $\quad$ $v \times 1$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern
1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet {0,1}; state is last *m* = 3 characters

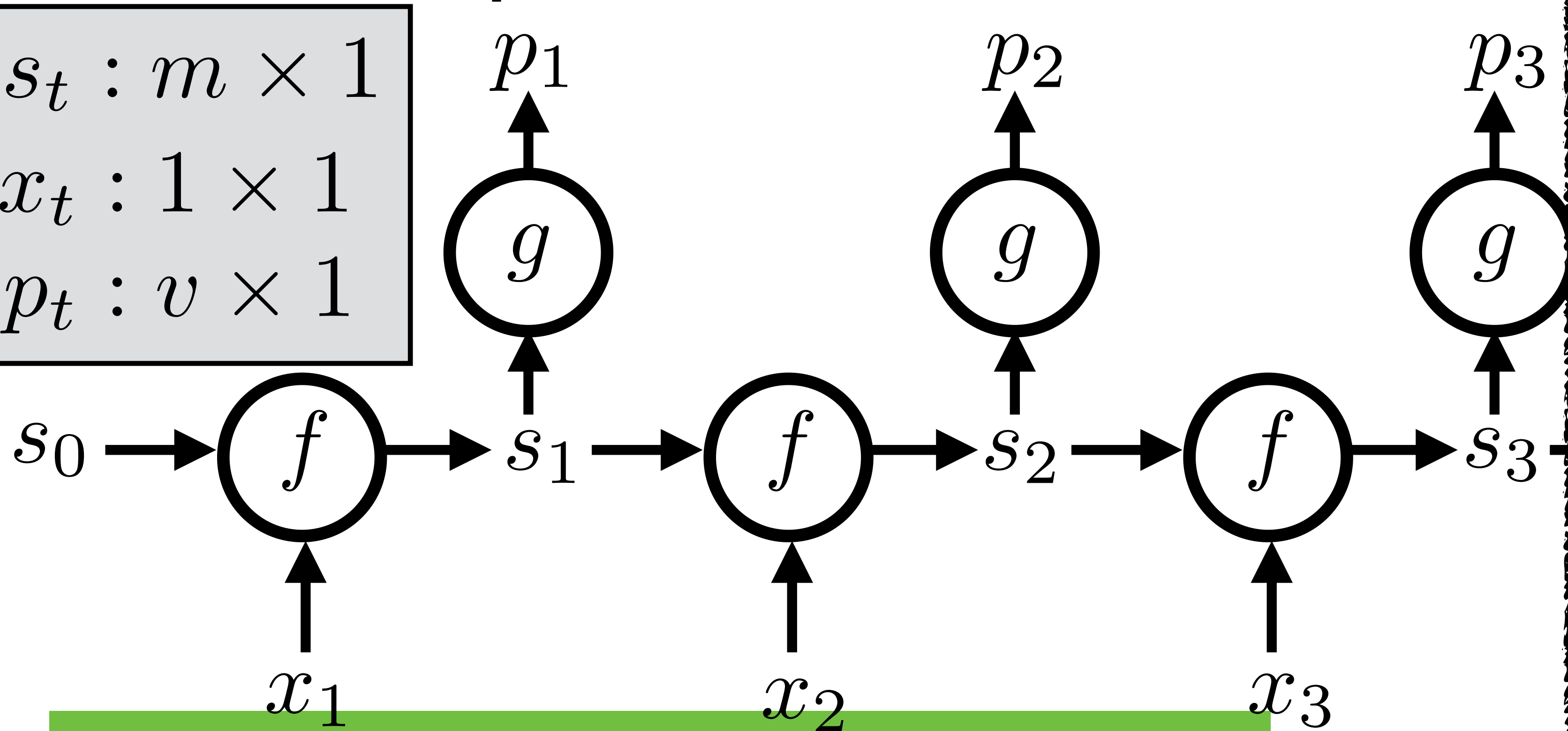
$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underset{v \times m}{W^o} s_t + \underset{v \times 1}{W_0^o}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



- Example: Alphabet {0,1}; state is last *m* = 3 characters

Recall: familiar pattern
1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

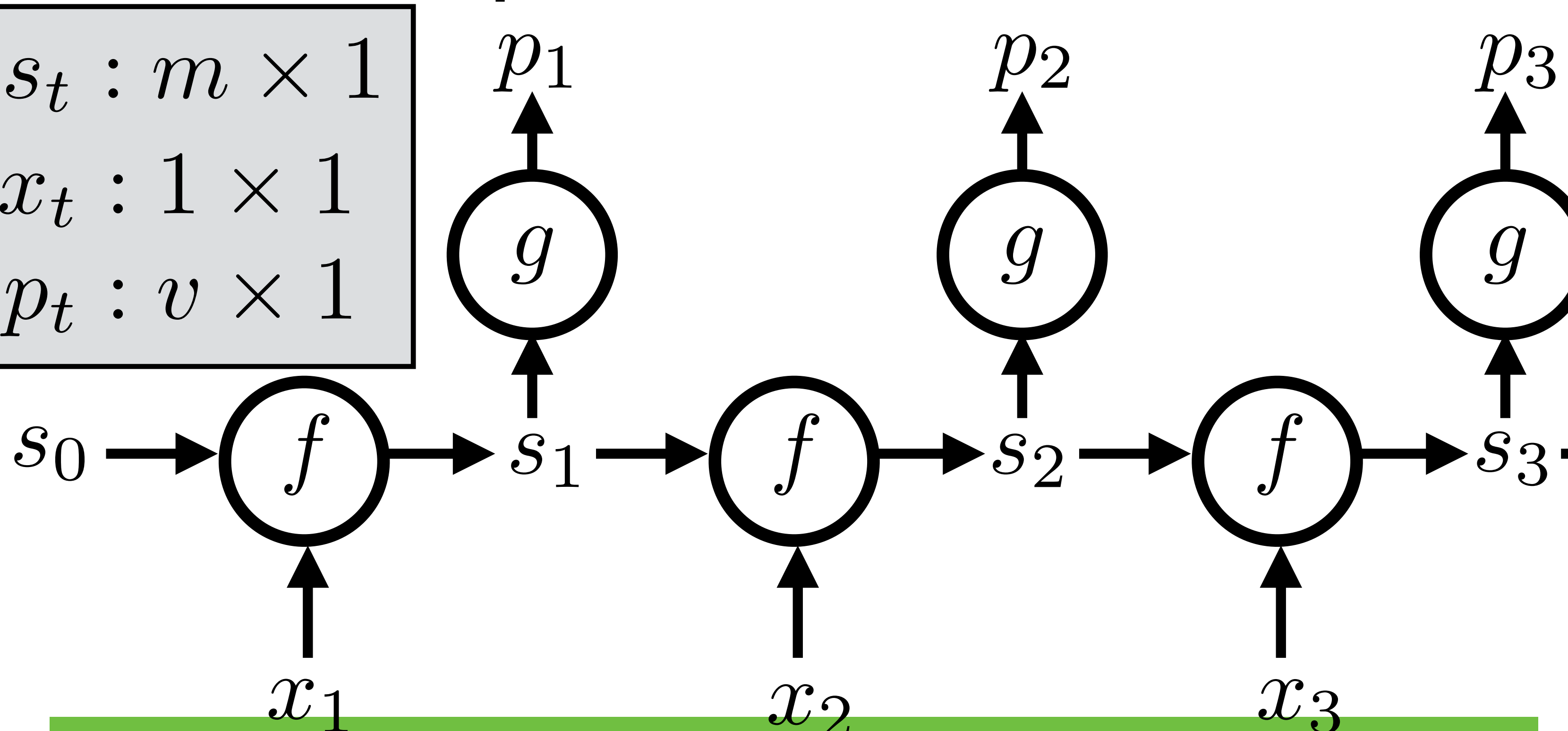
$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : 1 \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet of ℓ characters; state is last *c* characters

$$s_t = f_1\left(W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

$v \times m$ $\quad v \times 1$

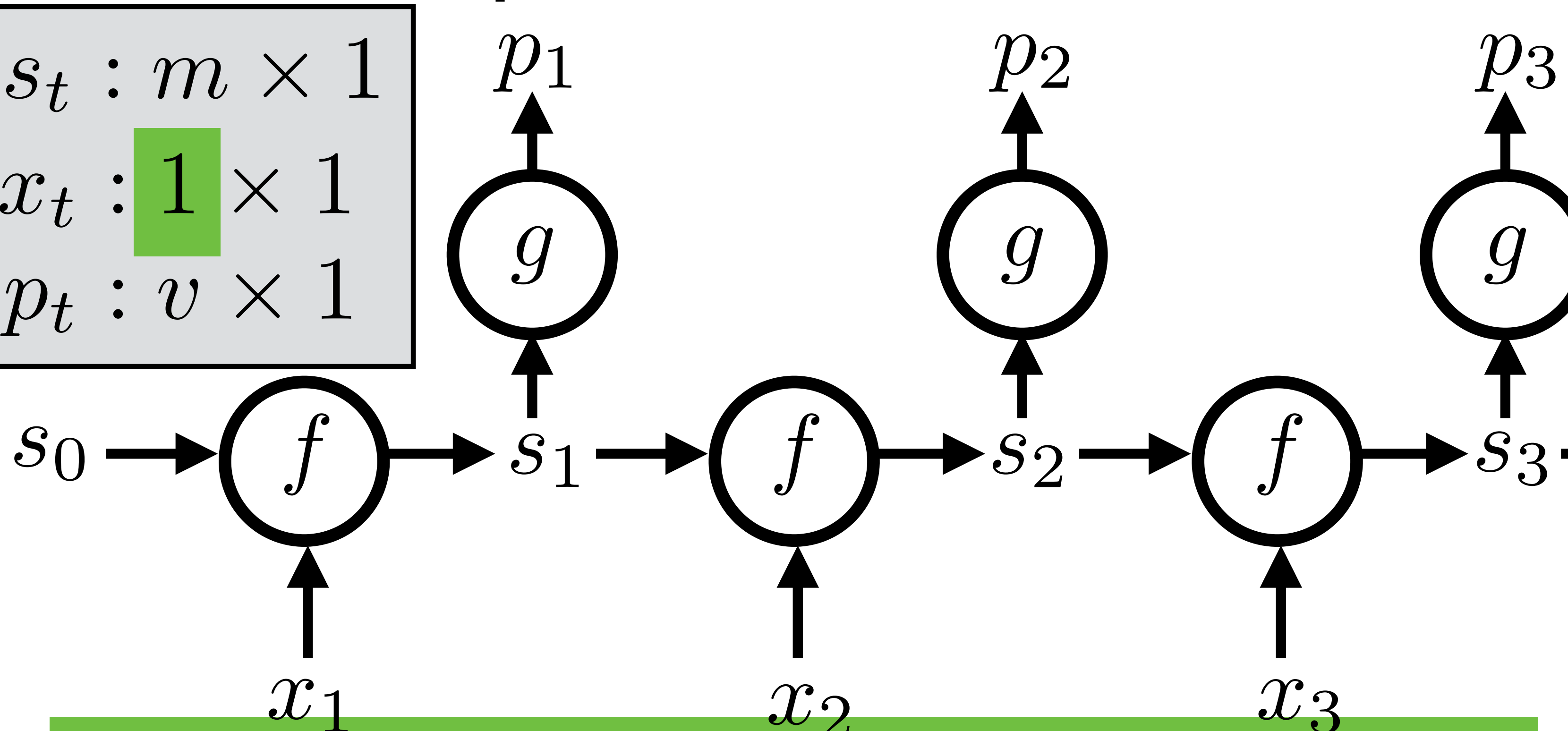
$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$

$x_t : \boxed{1} \times 1$

$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet of ℓ characters; state is last *c* characters

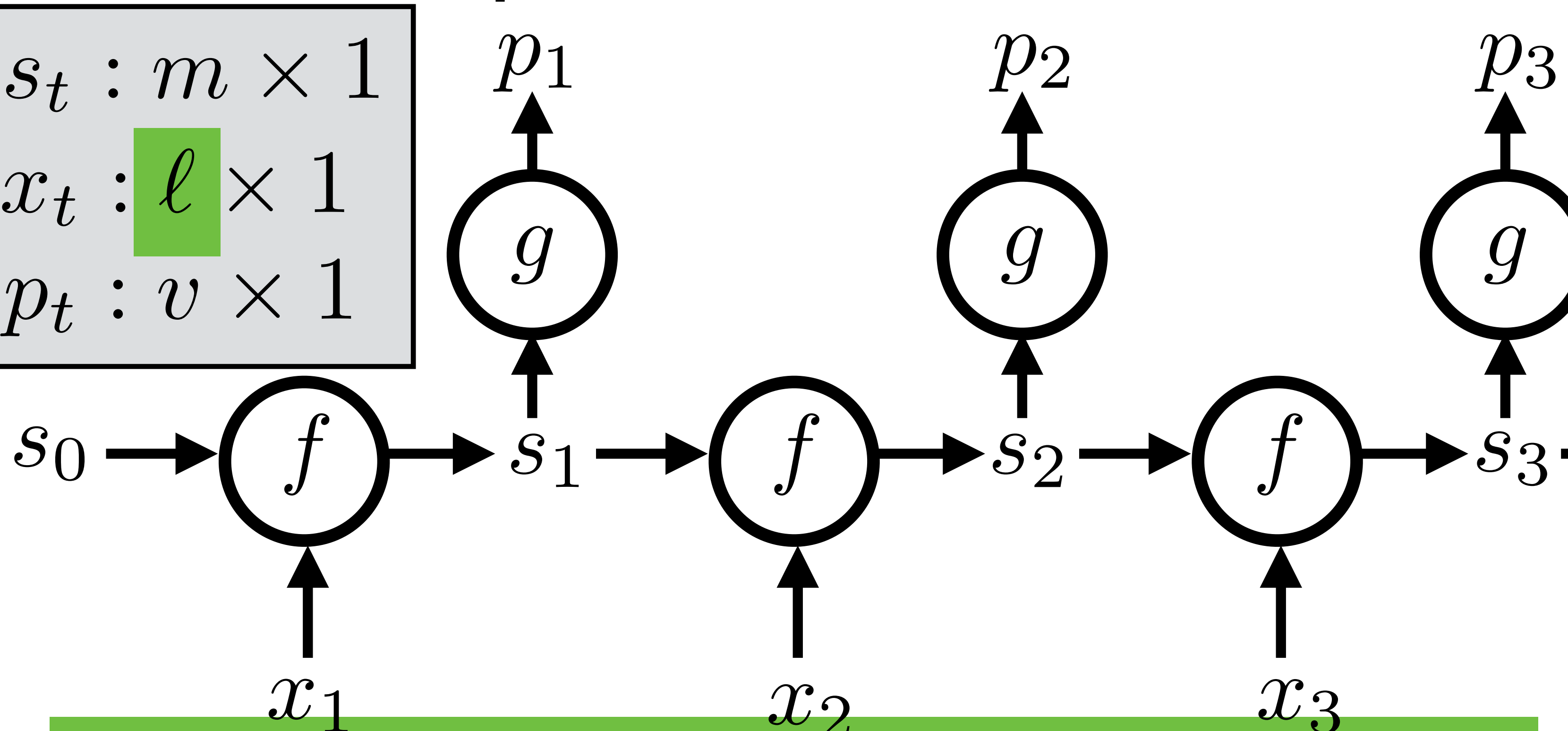
$$s_t = f_1\left(W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet of ℓ characters; state is last *c* characters

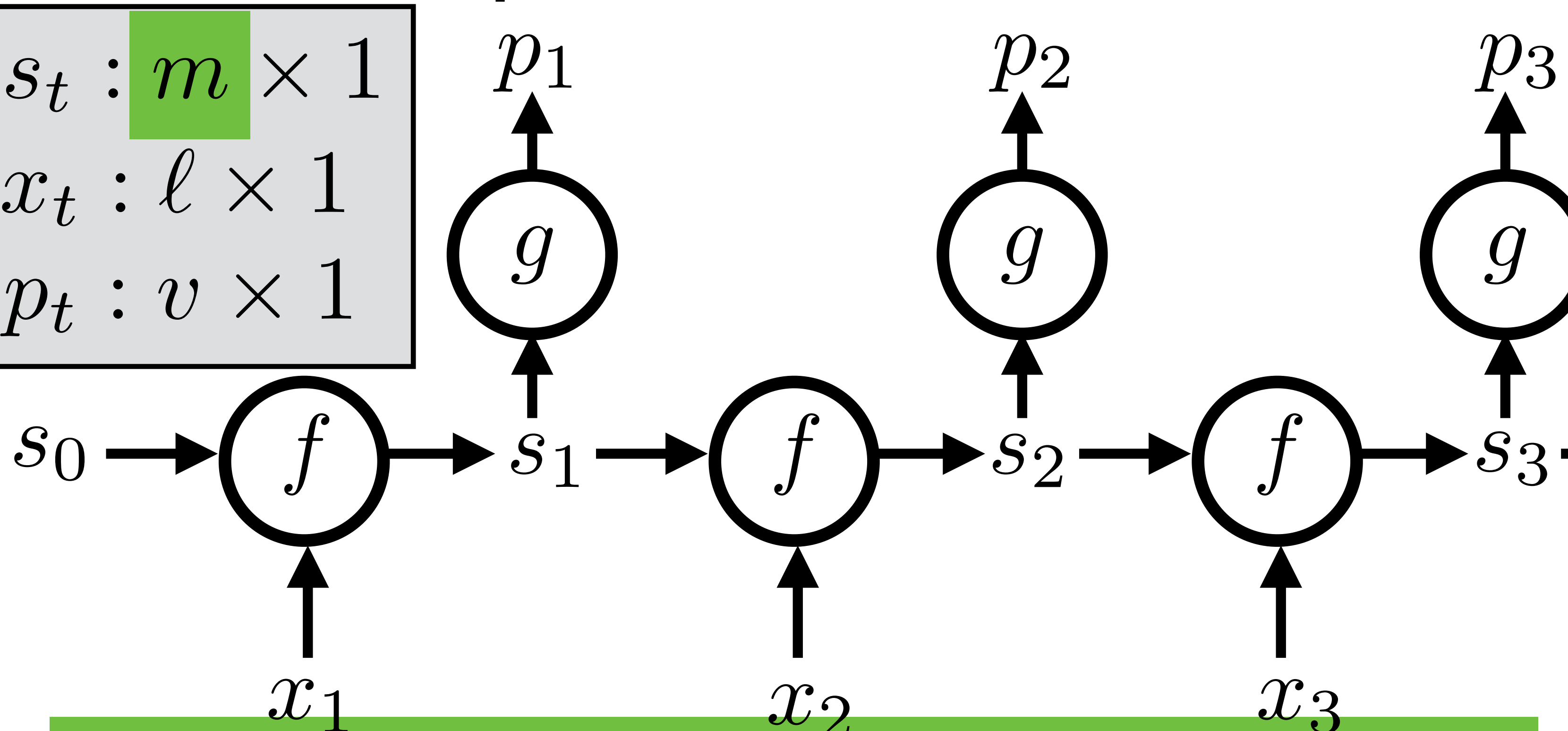
$$s_t = f_1\left(W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



- Example: Alphabet of ℓ characters; state is last c characters

Recall: familiar pattern
1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

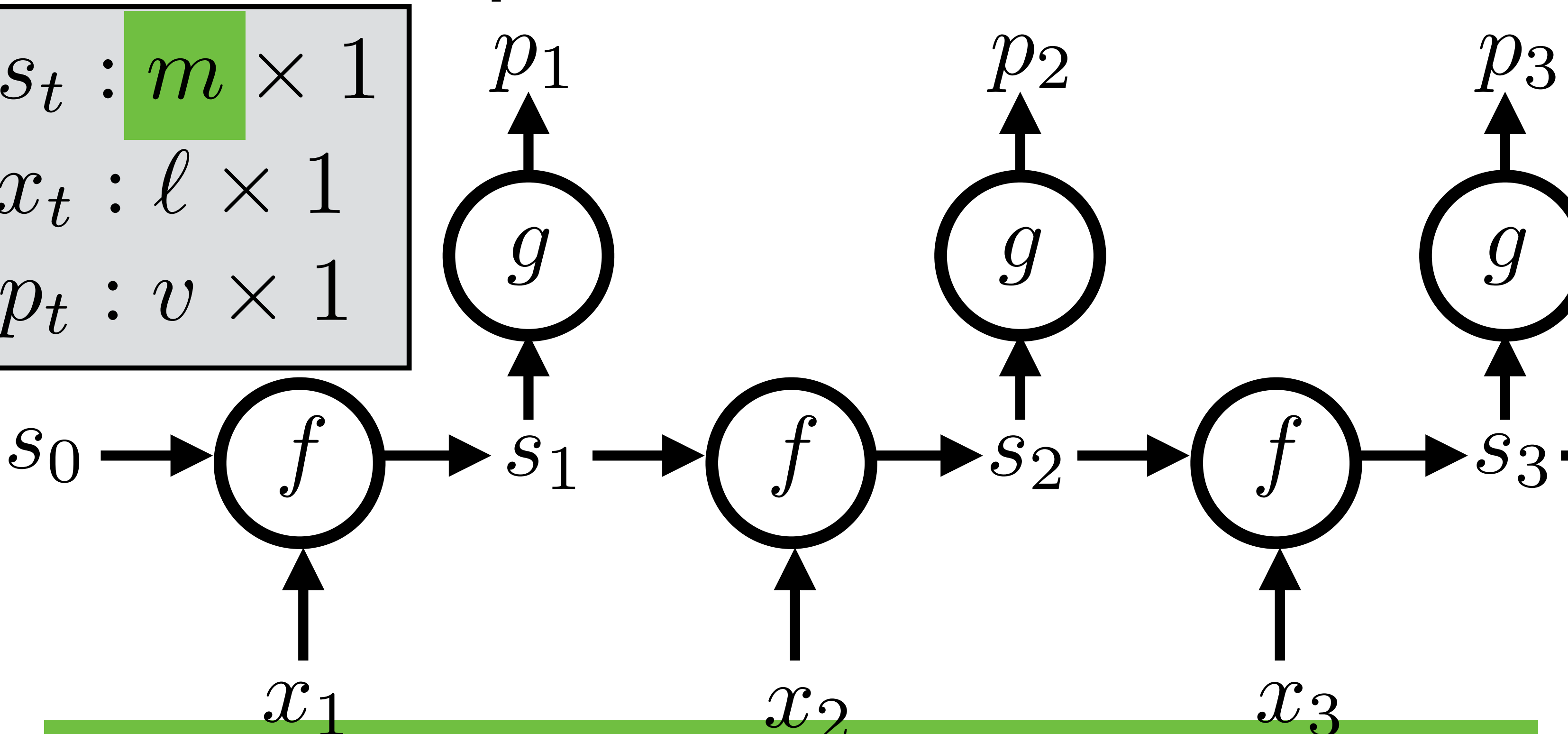
$$s_t = f_1\left(W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



Recall: familiar pattern
1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet of ℓ characters; state is last c characters ($m = c\ell$)

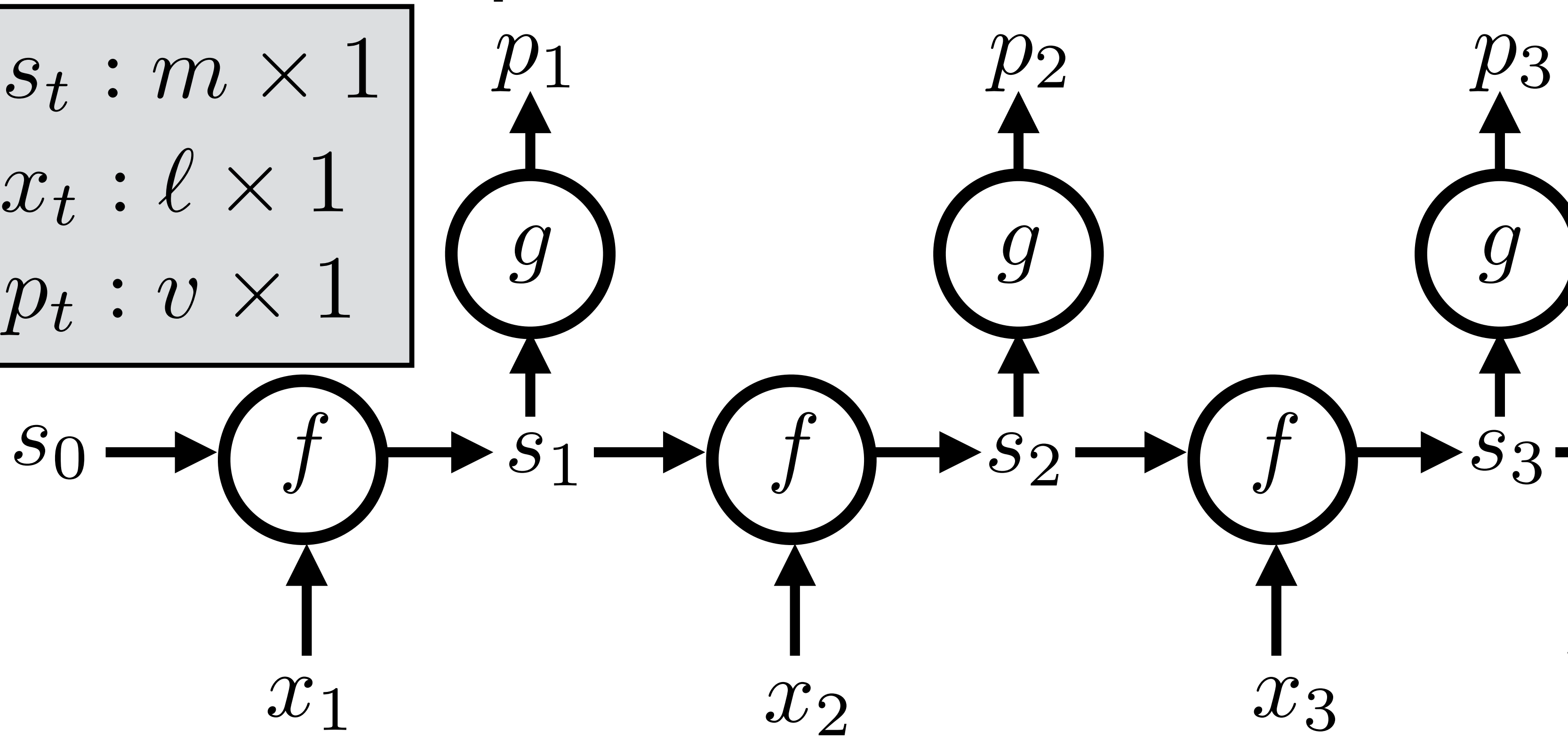
$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underset{v \times m}{W^o} s_t + \underset{v \times 1}{W_0^o}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$$s_t : m \times 1$$
$$x_t : \ell \times 1$$
$$p_t : v \times 1$$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet of ℓ characters; state is last *c* characters ($m = c\ell$)

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

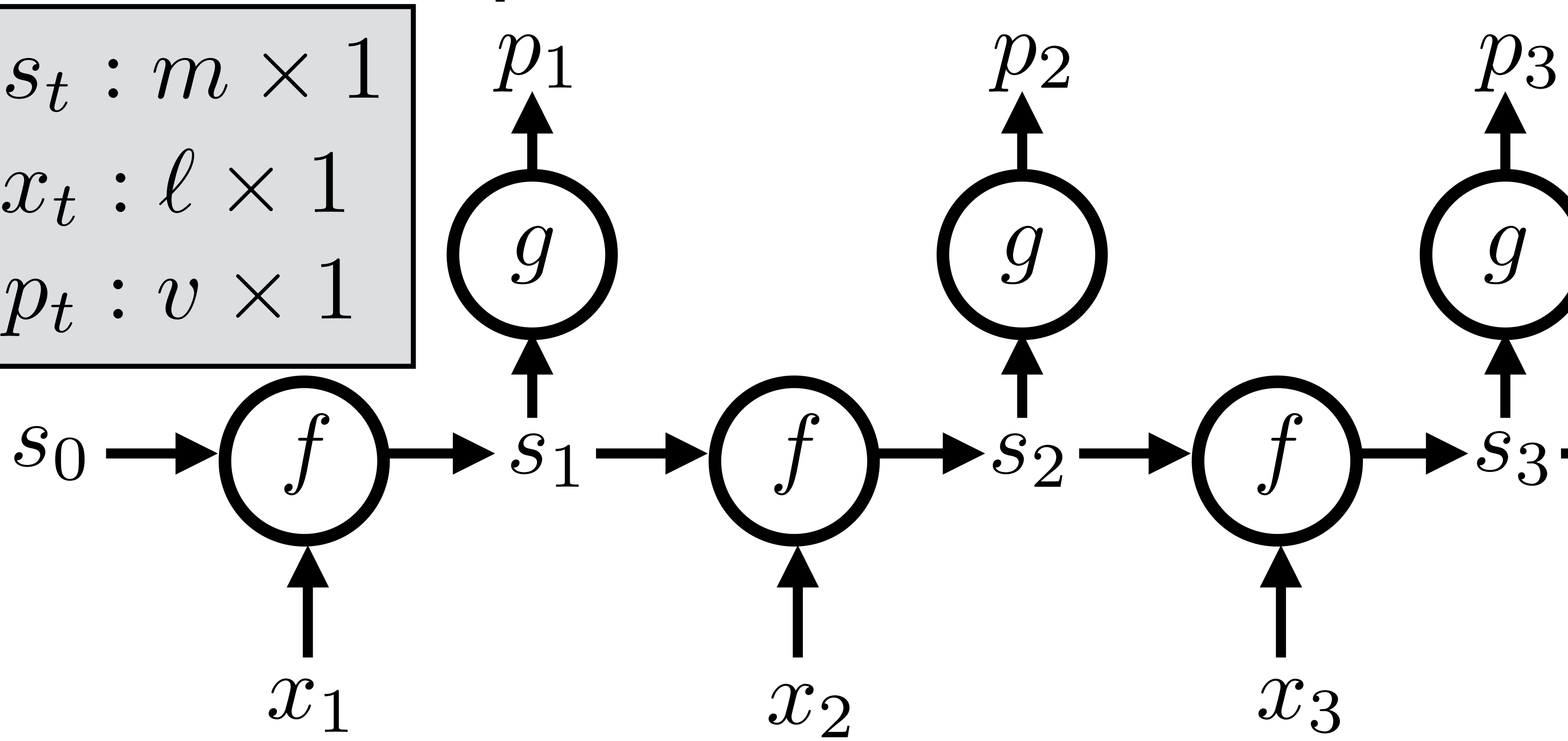
$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$

$x_t : \ell \times 1$

$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet of ℓ characters; state is last *c* characters ($m = c\ell$)

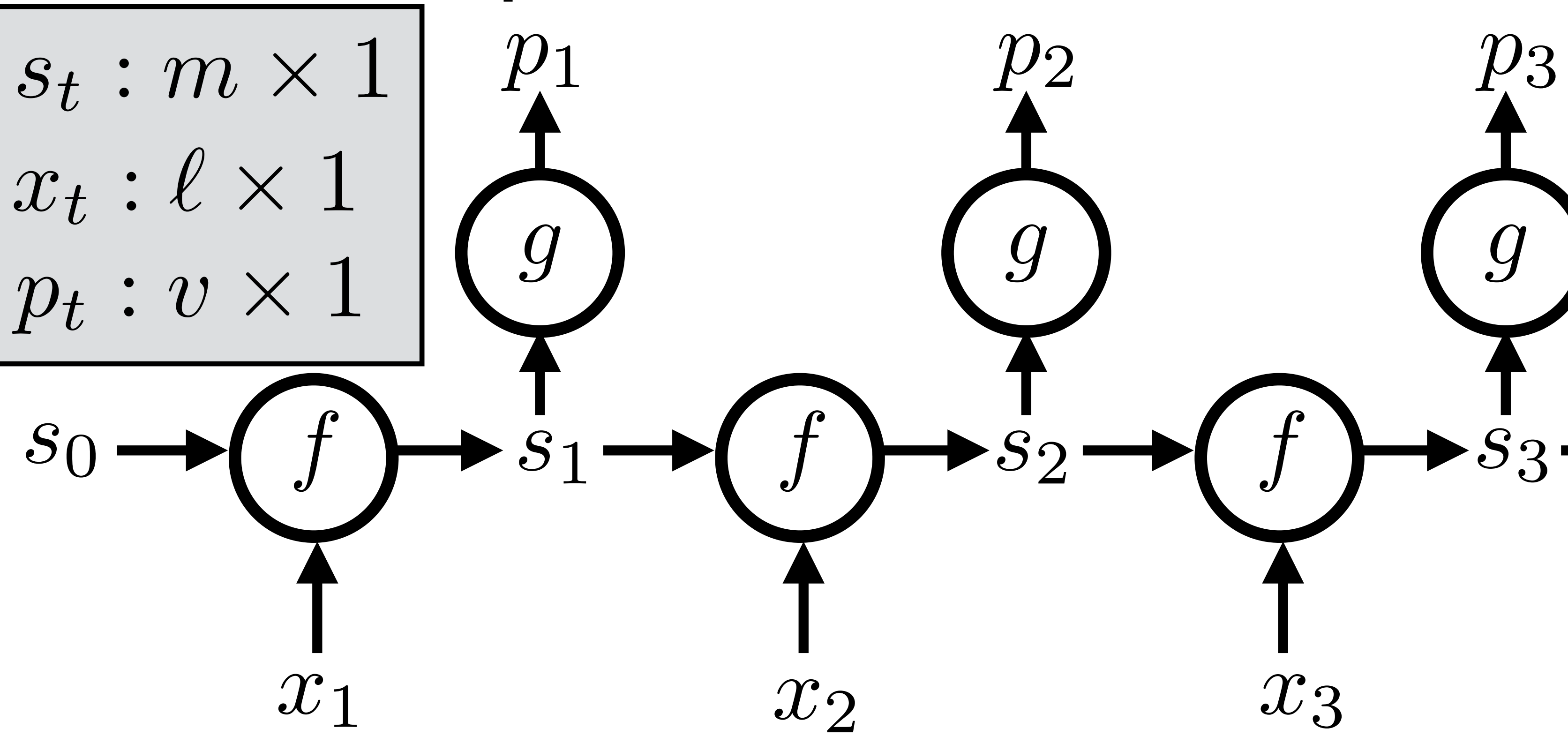
$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{W^o}_{v \times m} s_t + \underbrace{W_0^o}_{v \times 1}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet of ℓ characters; state is last *c* characters ($m = c\ell$)

$$s_t = f_1 \left(\boxed{W^{sx}} x_t + \boxed{W^{ss}} s_{t-1} + \boxed{W_0^{ss}} \right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underbrace{\boxed{W^o}}_{v \times m} s_t + \underbrace{\boxed{W_0^o}}_{v \times 1}) \end{aligned}$$

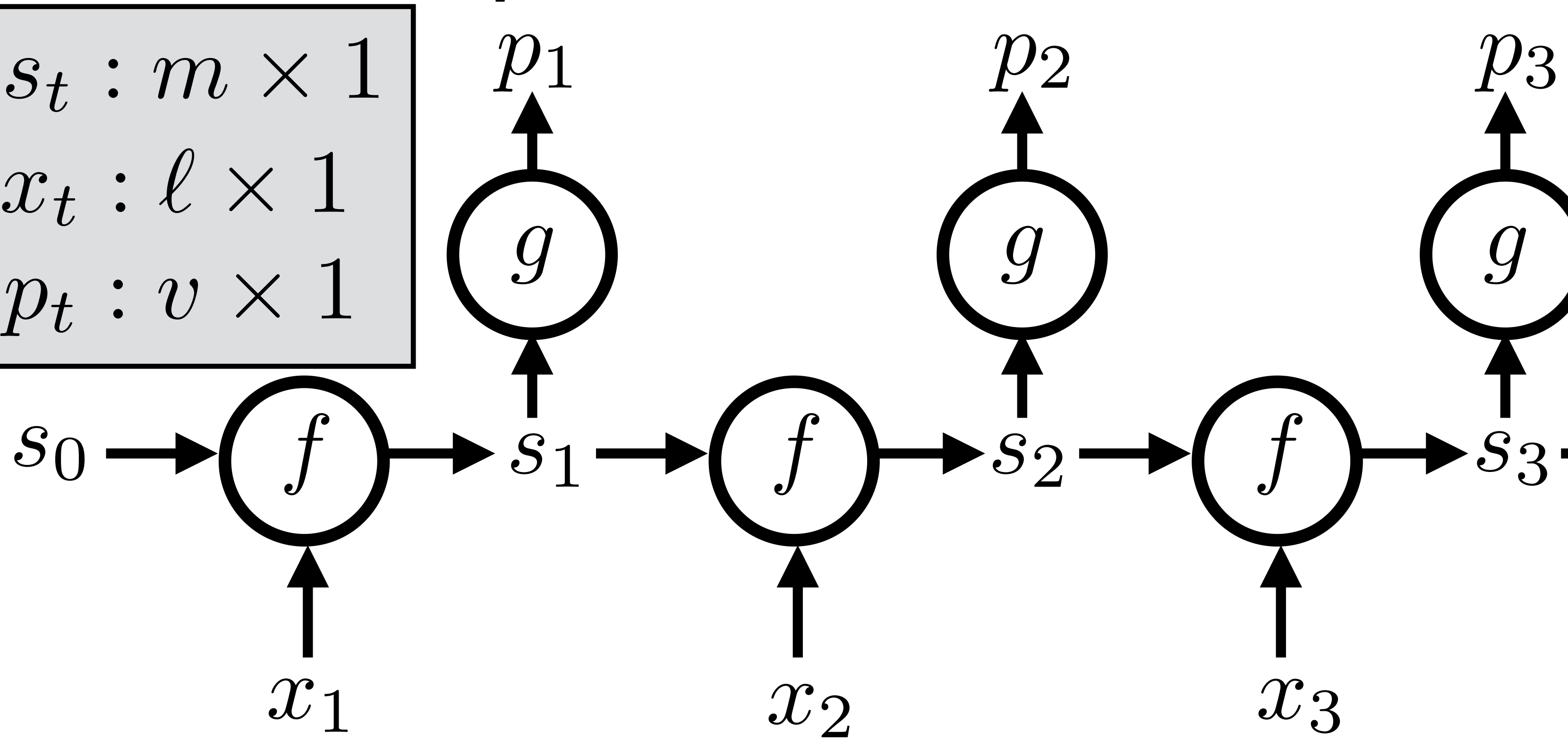
$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$

$x_t : \ell \times 1$

$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet of ℓ characters; state is last *c* characters ($m = c\ell$)

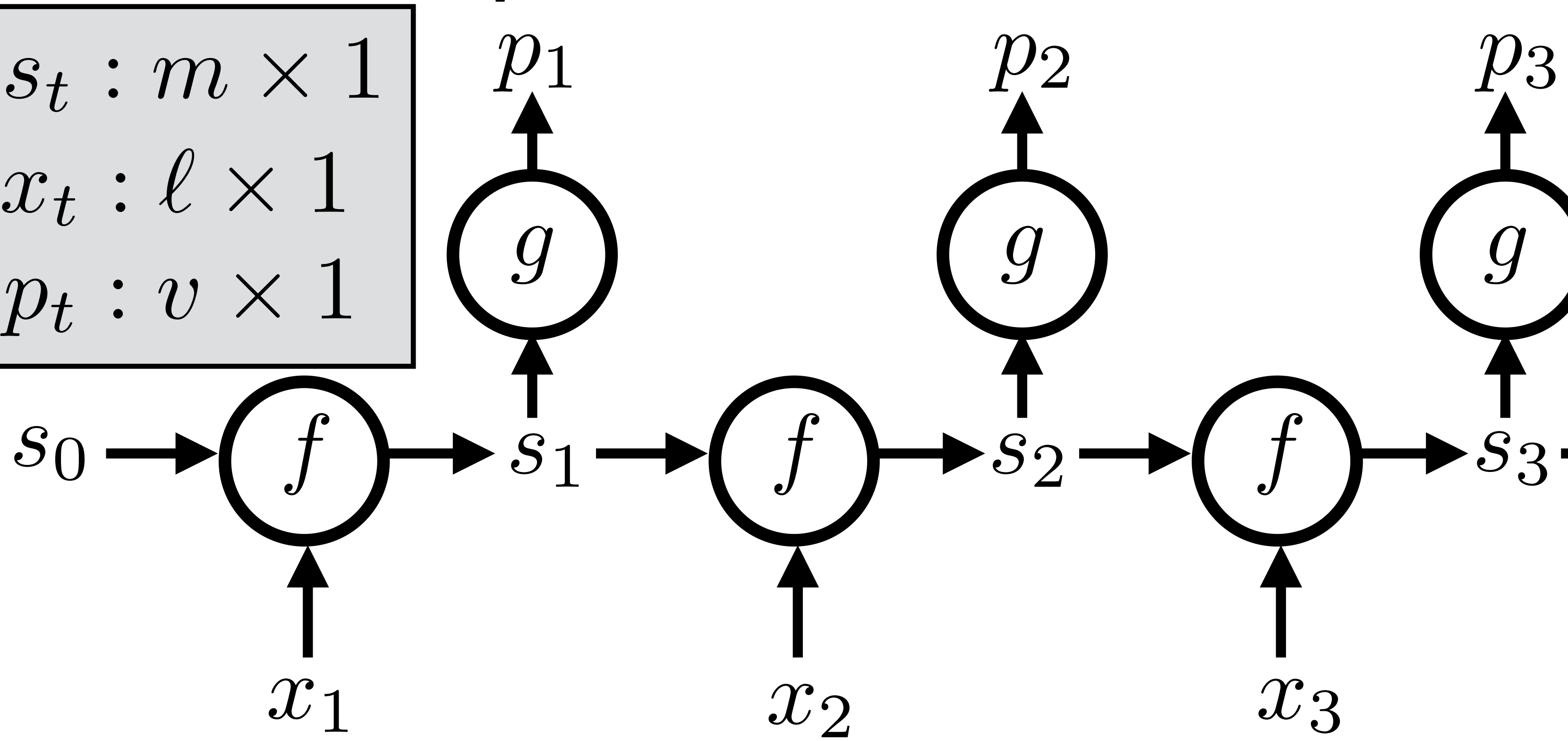
$$s_t = f_1 \left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss} \right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(\underset{v \times m}{W^o} s_t + \underset{v \times 1}{W_0^o}) \end{aligned}$$

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \mathrm{R}(x^{(i)}; W^o, W_0^o) \\ J(W^o, W_0^o) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Can express as a state machine

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet of ℓ characters; state is last *c* characters ($m = c\ell$)

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

$v \times m$ $\quad$ $v \times 1$

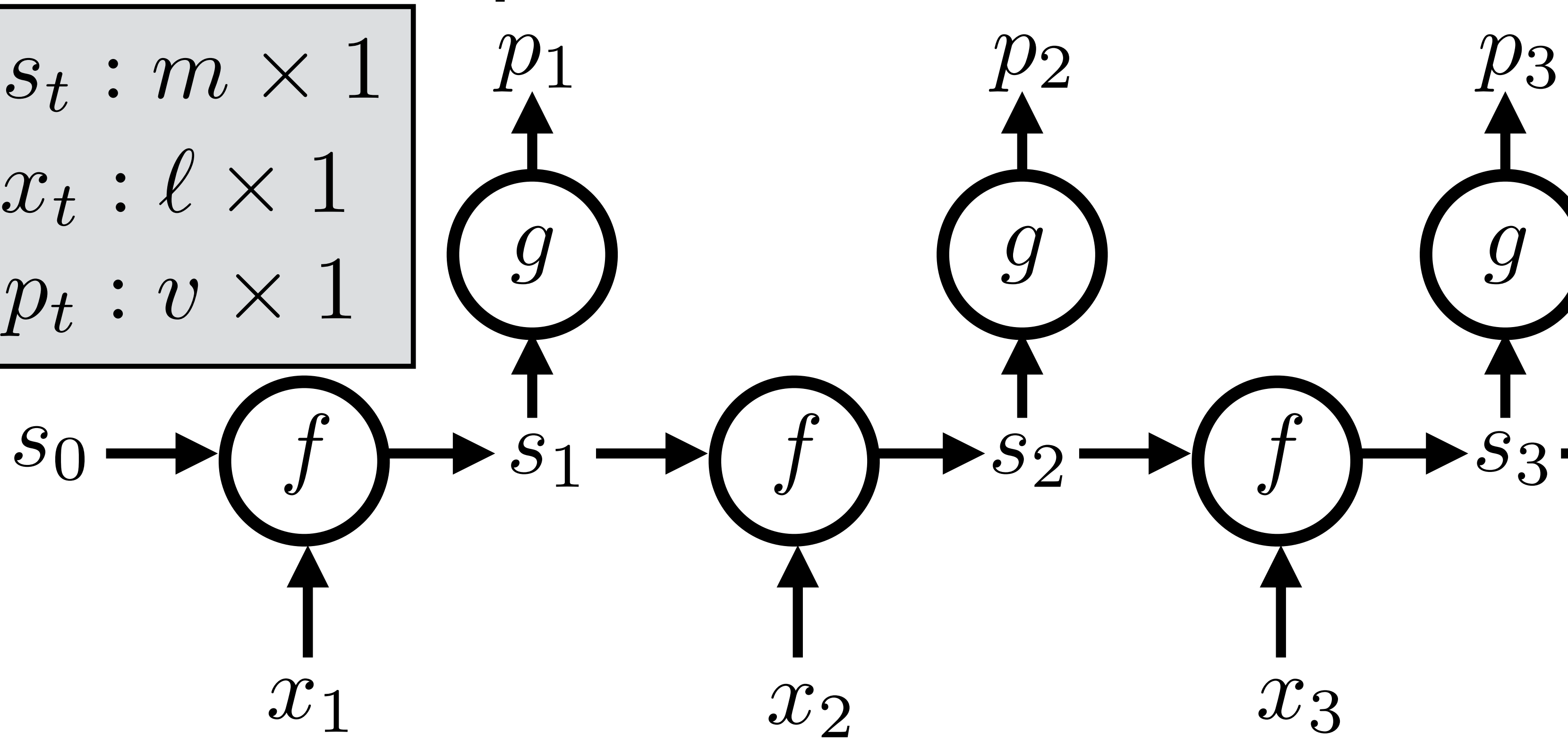
$$L_{\text{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$

$$p^{(i)} = \mathrm{R}(x^{(i)}; W, W_0)$$

$$J(W, W_0) = \sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right)$$

Can express as a state machine

$$s_t : m \times 1$$
$$x_t : \ell \times 1$$
$$p_t : v \times 1$$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet of ℓ characters; state is last *c* characters ($m = c\ell$)

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

v x *m*   v x 1

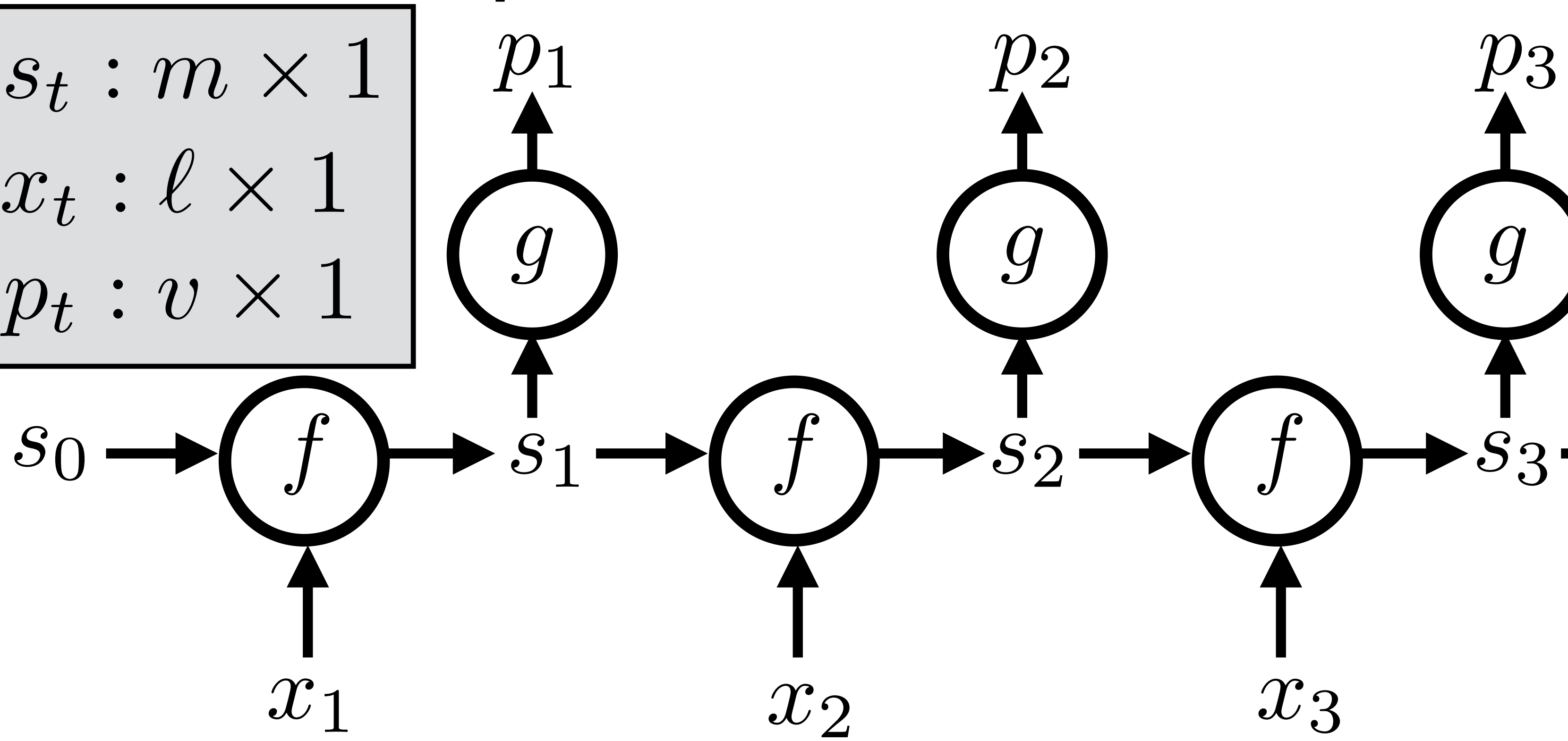
$$L_{\text{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$
$$p^{(i)} = \mathrm{R}(x^{(i)}; W, W_0)$$
$$J(W, W_0) = \sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right)$$

Can express as a state machine

$s_t : m \times 1$

$x_t : \ell \times 1$

$p_t : v \times 1$



Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

- Example: Alphabet of ℓ characters; state is last *c* characters ($m = c\ell$)

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

v x *m* *v* x 1

$$\begin{aligned} L_{\text{seq}}(p^{(i)}, y^{(i)}) &= \textstyle\sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)}) \\ p^{(i)} &= \text{RNN}(x^{(i)}; W, W_0) \\ J(W, W_0) &= \textstyle\sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right) \end{aligned}$$

Recurrent neural network

Recurrent neural network

- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

Recurrent neural network

- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

$$s_t = f_1\left(W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}\right)$$

$$p_t = f_2(W^o s_t + W_0^o)$$

Recurrent neural network

- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$p_t = f_2(W^o s_t + W_0^o)$$

$$p^{(i)} = \text{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$$s_t : m \times 1$$
$$x_t : \ell \times 1$$
$$p_t : v \times 1$$

- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$p_t = f_2(W^o s_t + W_0^o)$$

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$

s_0

x_1

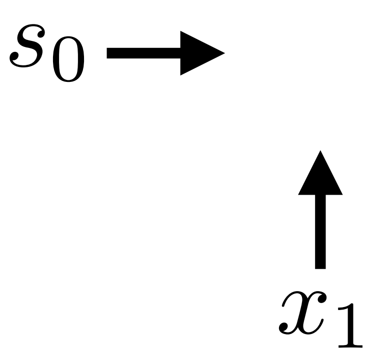
- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

$$s_t = f_1 \left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss} \right)$$

$$p_t = f_2(W^o s_t + W_0^o) \qquad p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$$\boxed{\begin{aligned} s_t &: m \times 1 \\ x_t &: \ell \times 1 \\ p_t &: v \times 1 \end{aligned}}$$



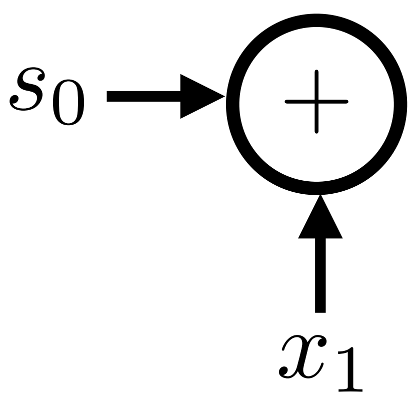
- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

$$s_t = f_1\left(W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}\right)$$

$$p_t = f_2(W^o s_t + W_0^o) \qquad\qquad p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



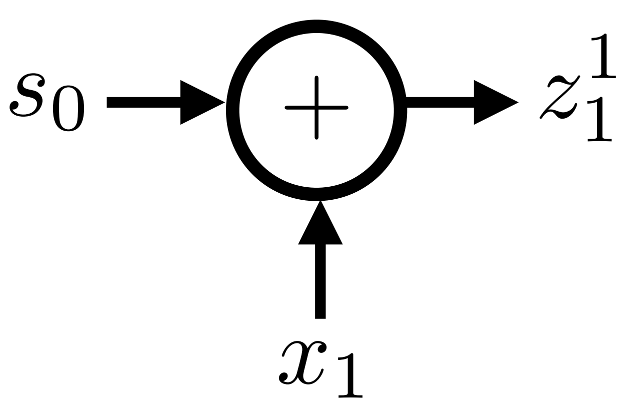
- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

$$s_t = f_1 \left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss} \right)$$

$$p_t = f_2(W^o s_t + W_0^o) \qquad\qquad p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



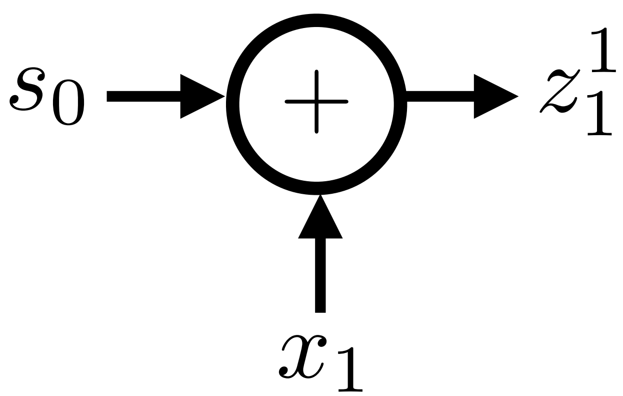
- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

$$s_t = f_1\left(W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}\right)$$

$$p_t = f_2(W^o s_t + W_0^o) \qquad p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



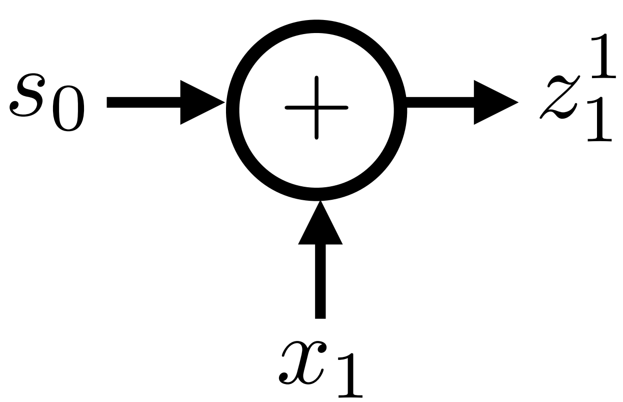
- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

$$s_t = f_1\left(W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}\right)$$

$$p_t = f_2(W^o s_t + W_0^o) \qquad p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



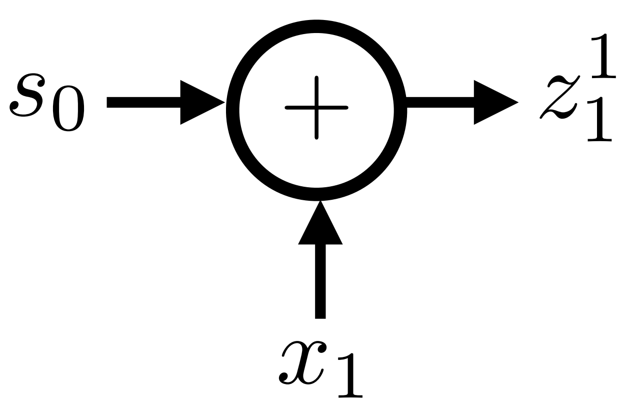
- Example: Alphabet of ℓ chars; state is last *c* chars $(m = c\ell)$

$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(W^o s_t + W_0^o) \qquad p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(W^o s_t + W_0^o)$$

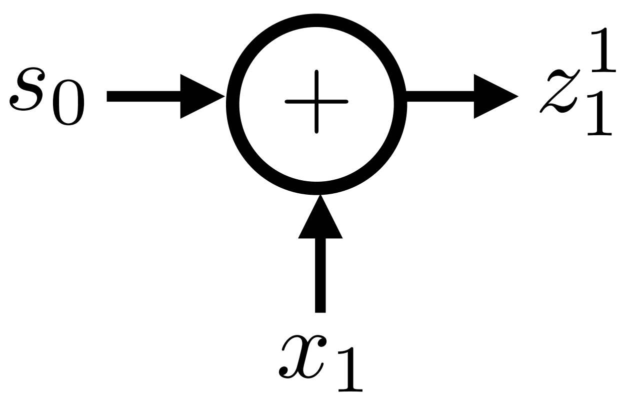
$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$

$x_t : \ell \times 1$

$p_t : v \times 1$



- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

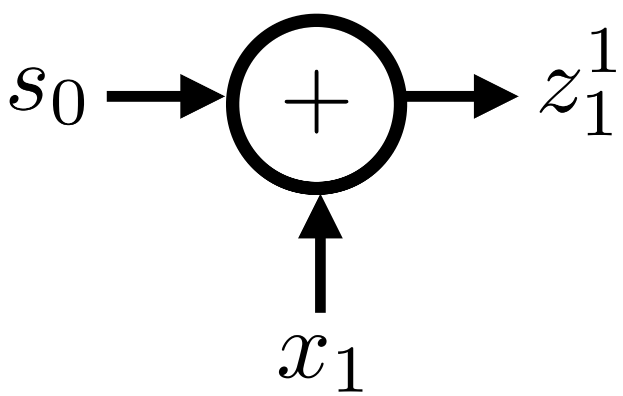
$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2\left(\underbrace{W^o s_t + W_0^o}_{z_t^2}\right)$$

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$$s_t : m \times 1$$
$$x_t : \ell \times 1$$
$$p_t : v \times 1$$



- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

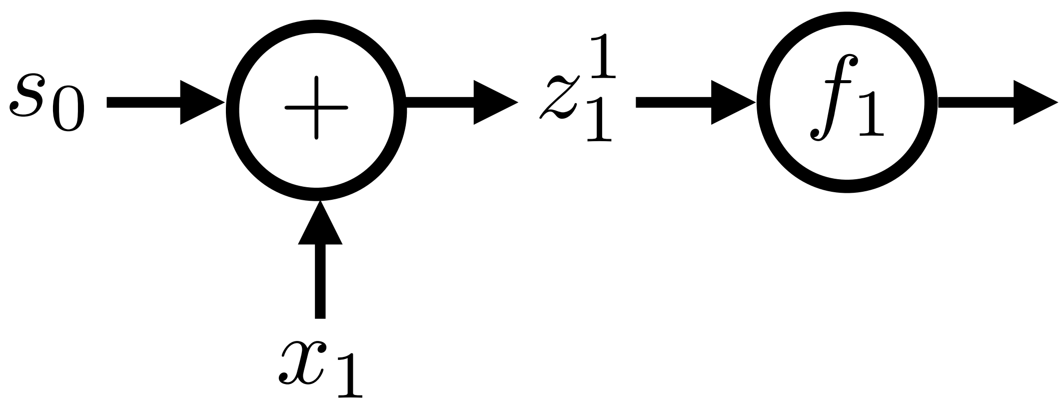
$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2})$$

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$$s_t : m \times 1$$
$$x_t : \ell \times 1$$
$$p_t : v \times 1$$



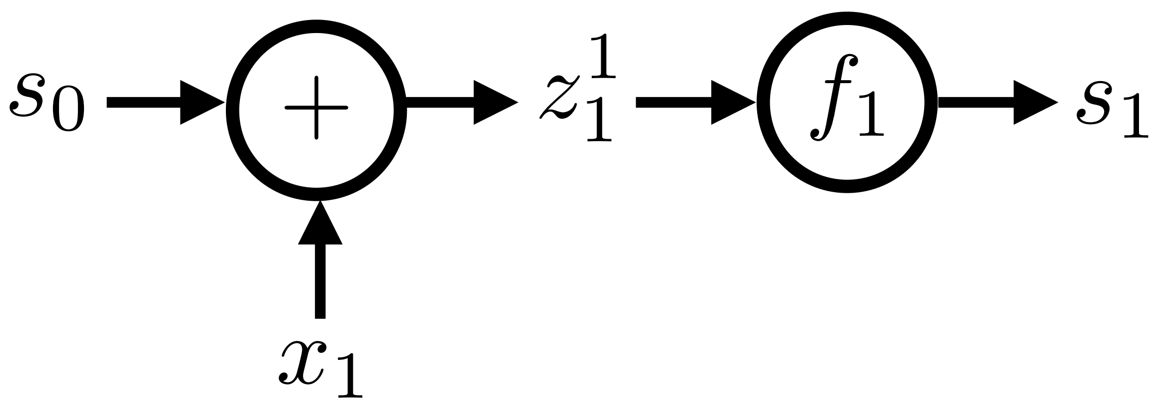
- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2}) \qquad p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

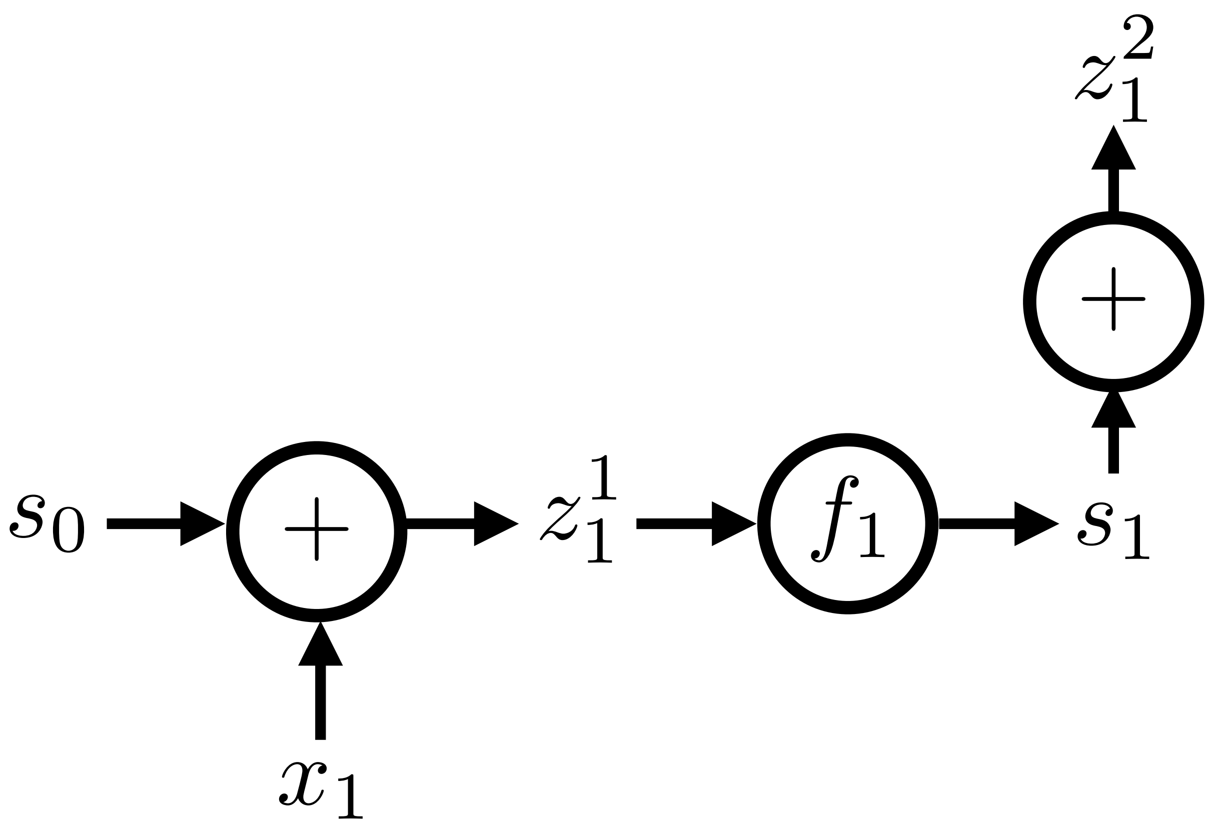
$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2})$$

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$$s_t : m \times 1$$
$$x_t : \ell \times 1$$
$$p_t : v \times 1$$



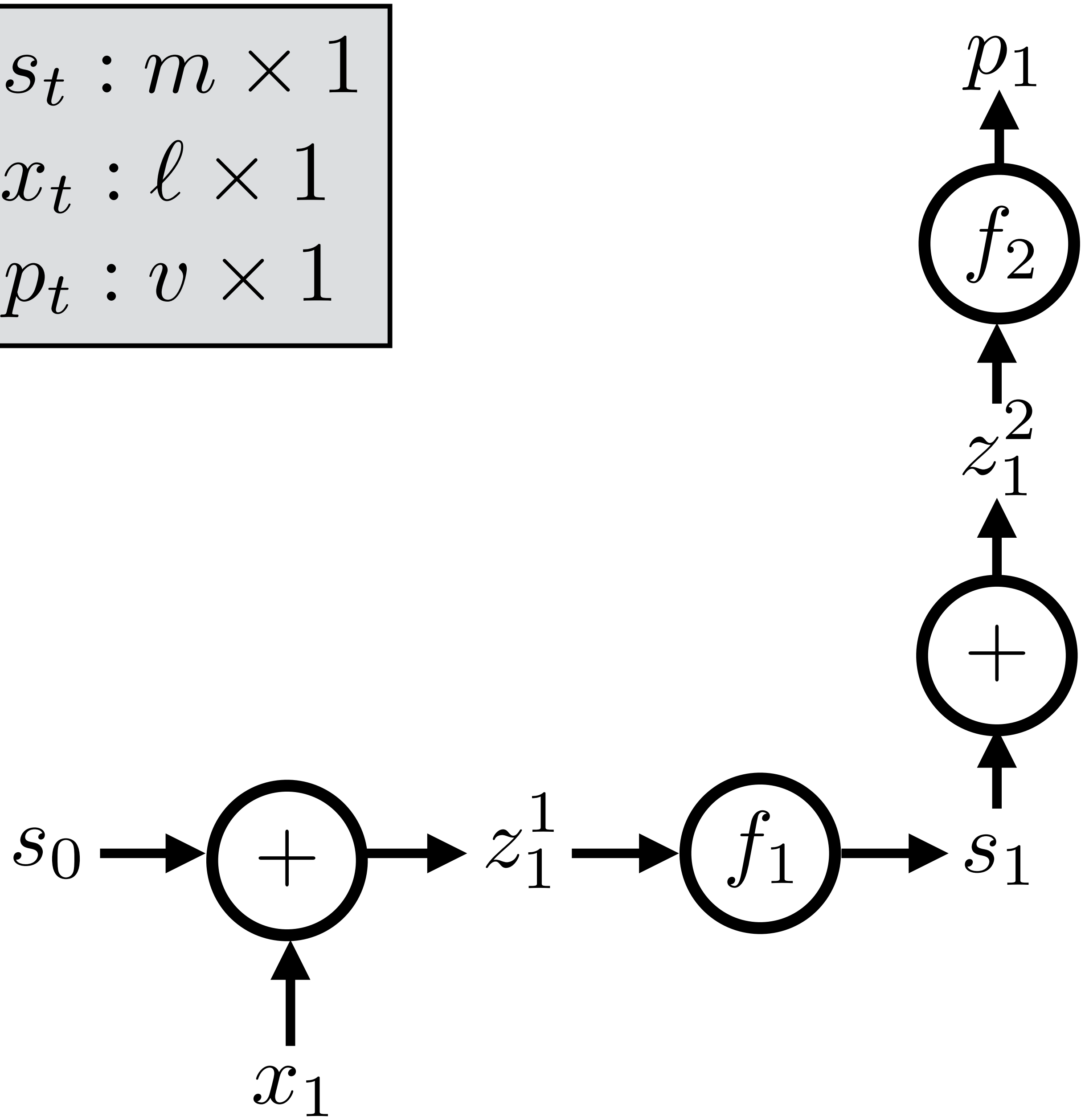
- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

$$s_t = f_1\,(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1})$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2}) \qquad\qquad p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2})$$

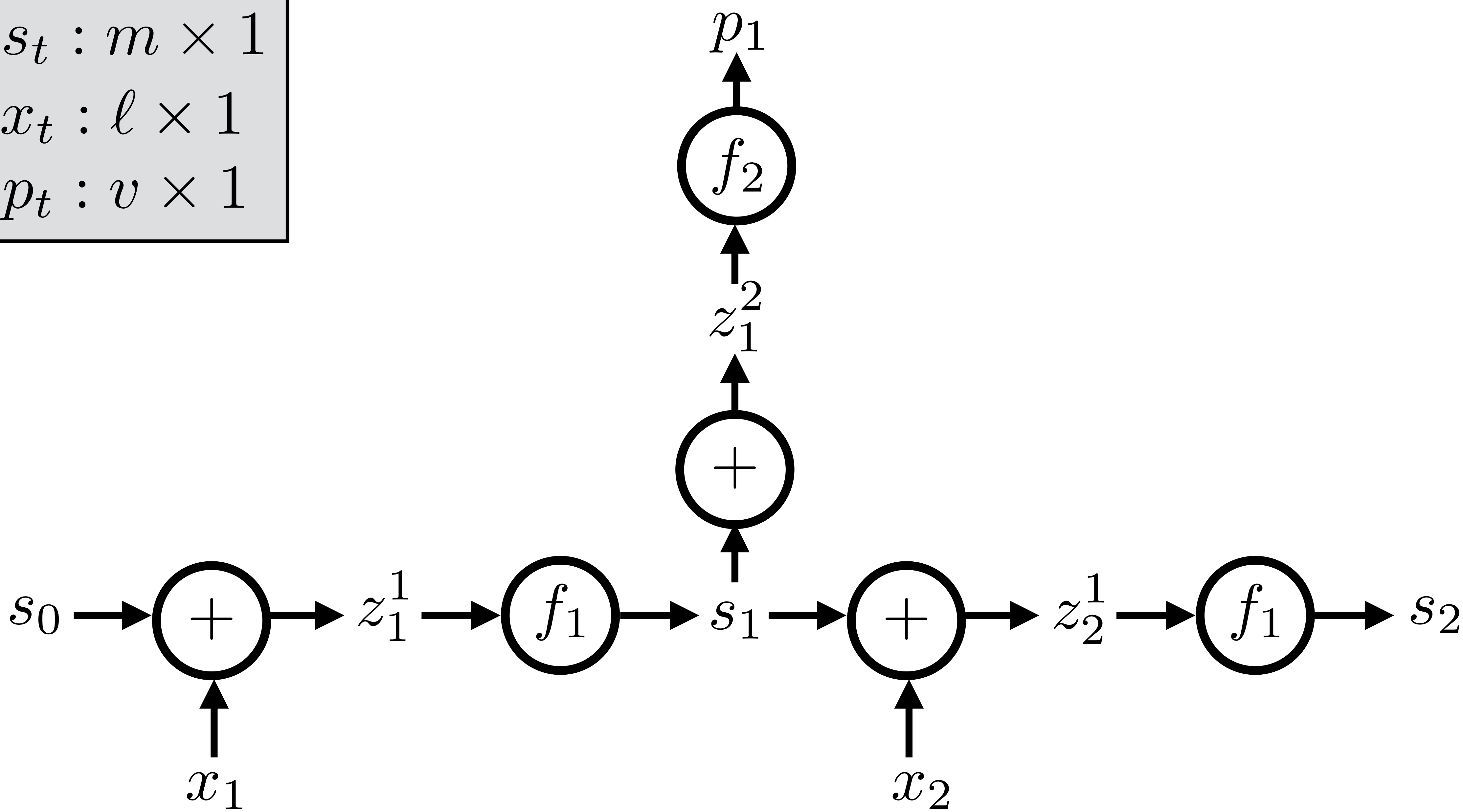
$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$

$x_t : \ell \times 1$

$p_t : v \times 1$



- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

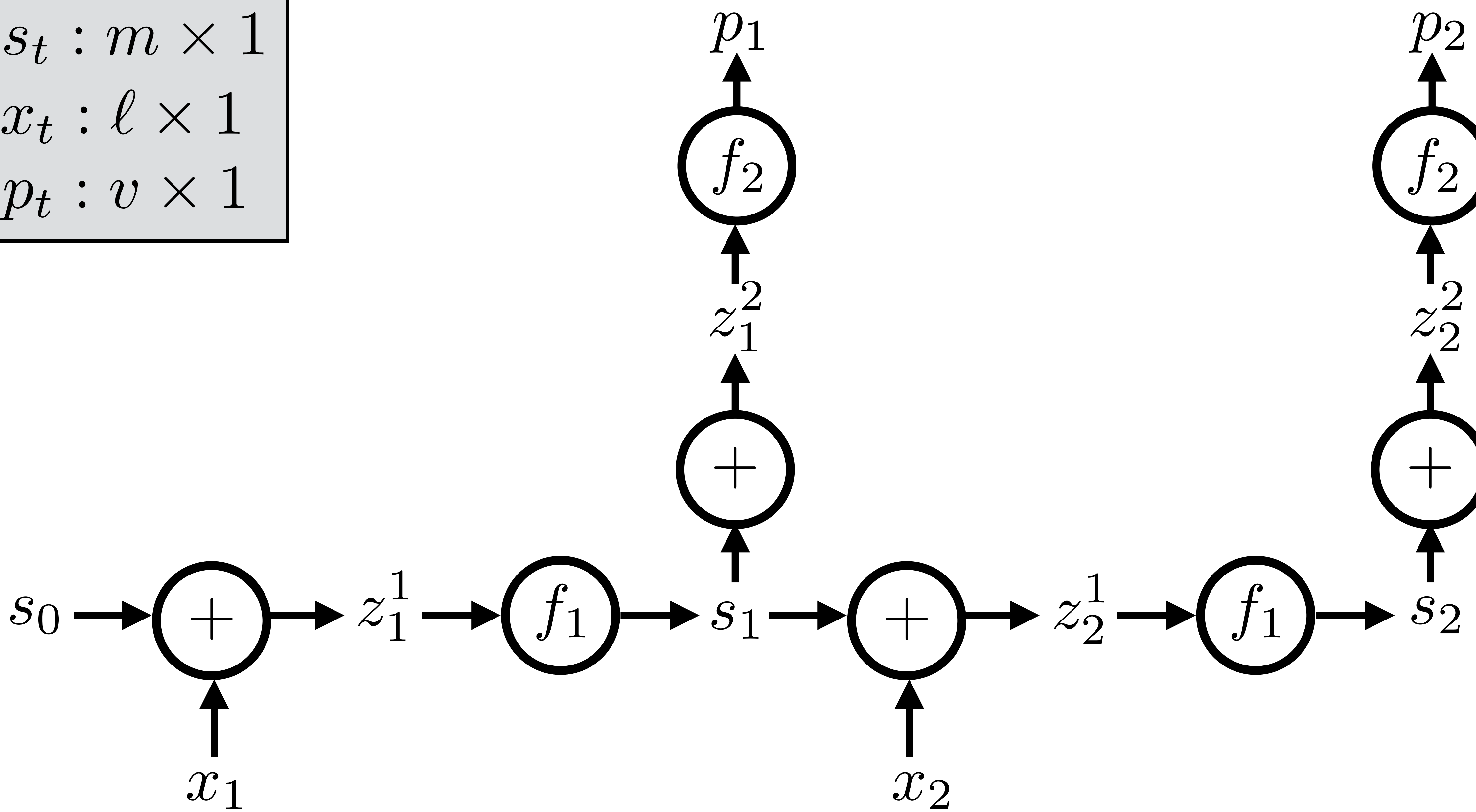
$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2})$$

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

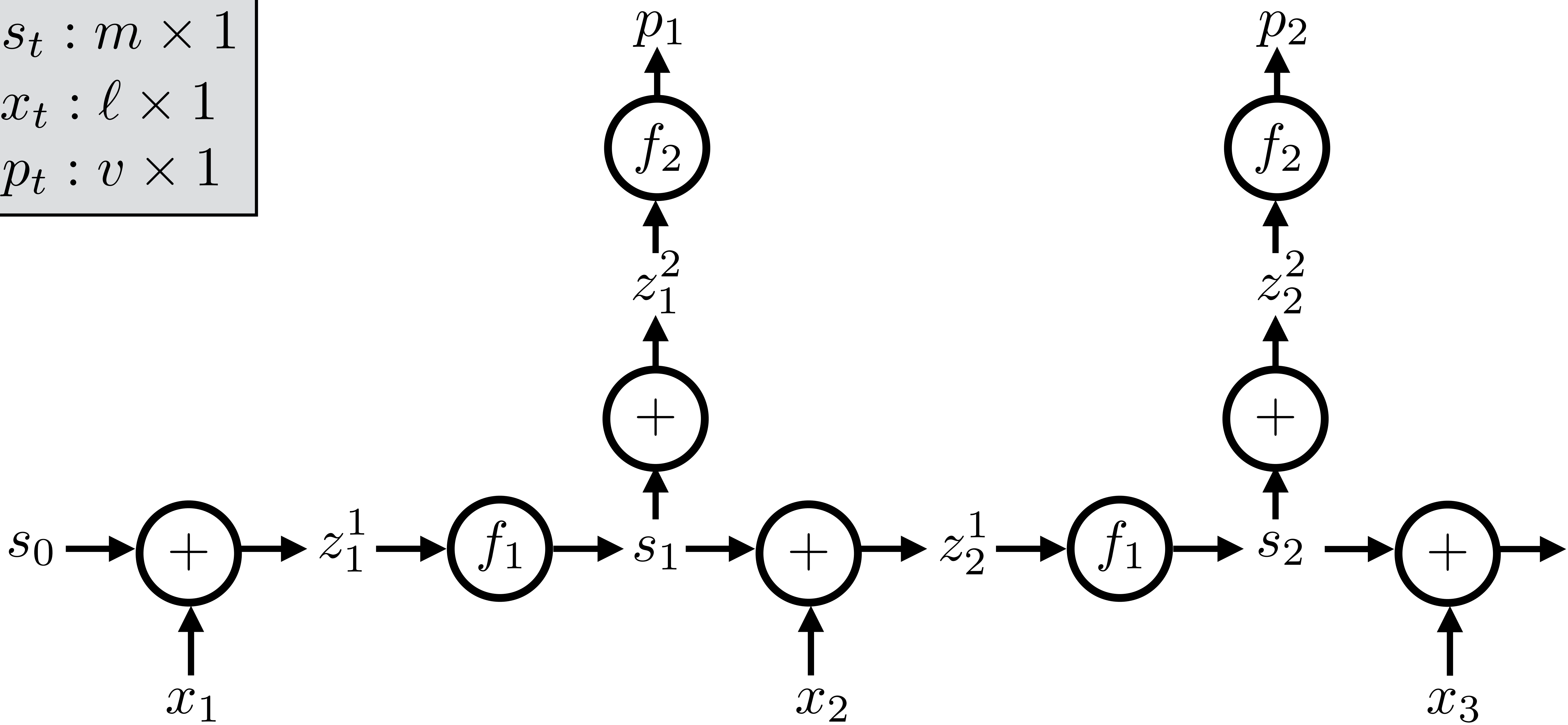
$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2})$$

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

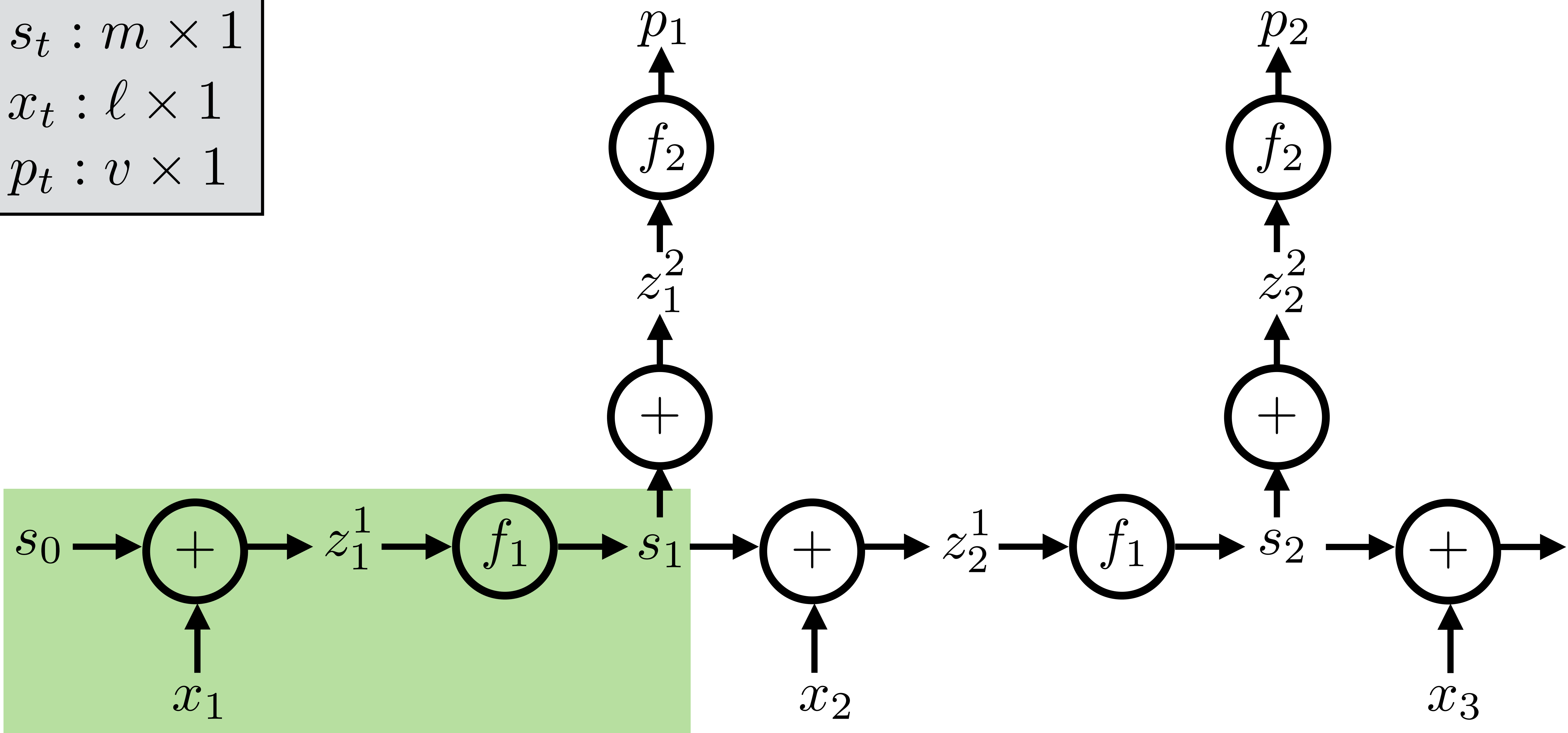
$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2})$$

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

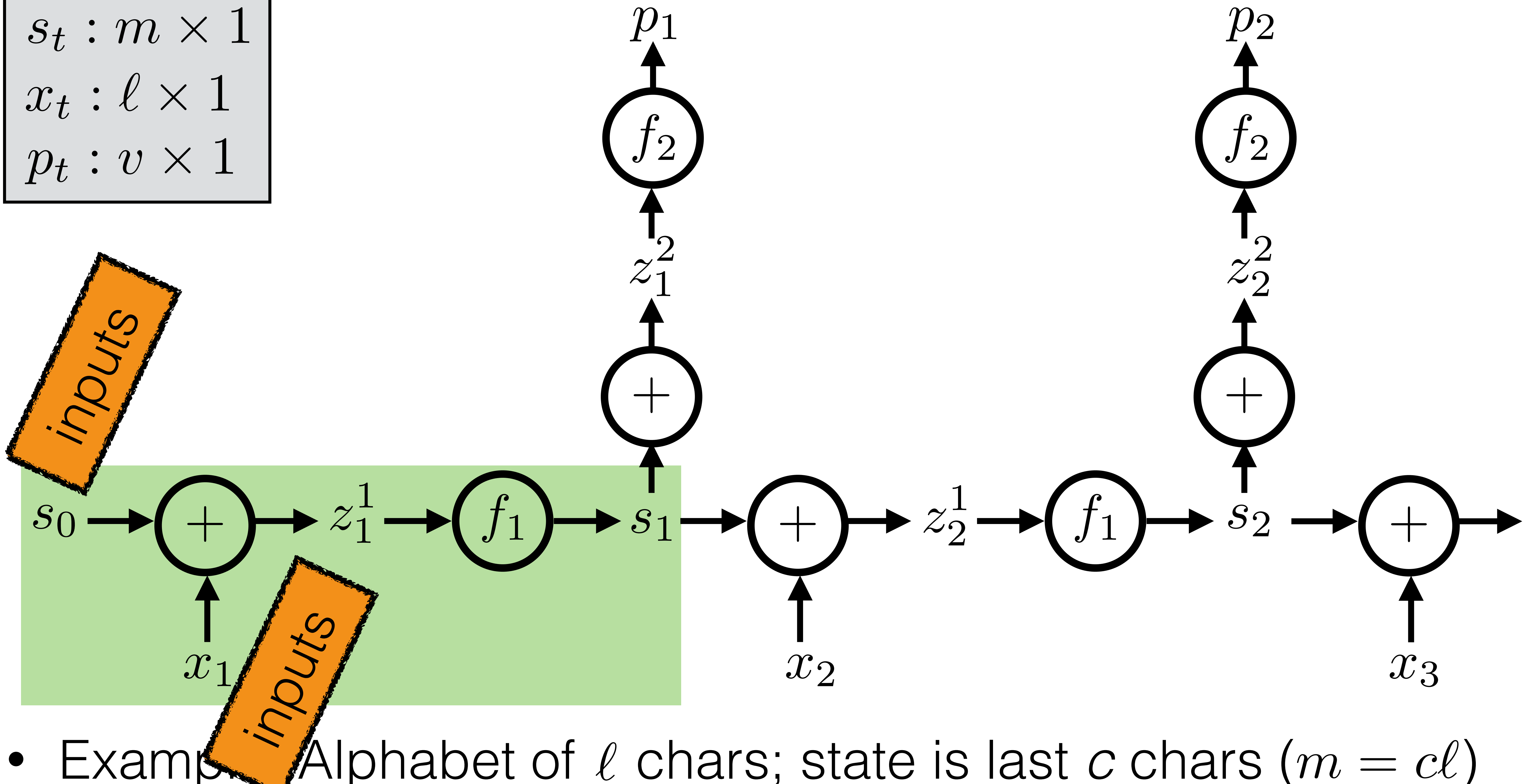
$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2})$$

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

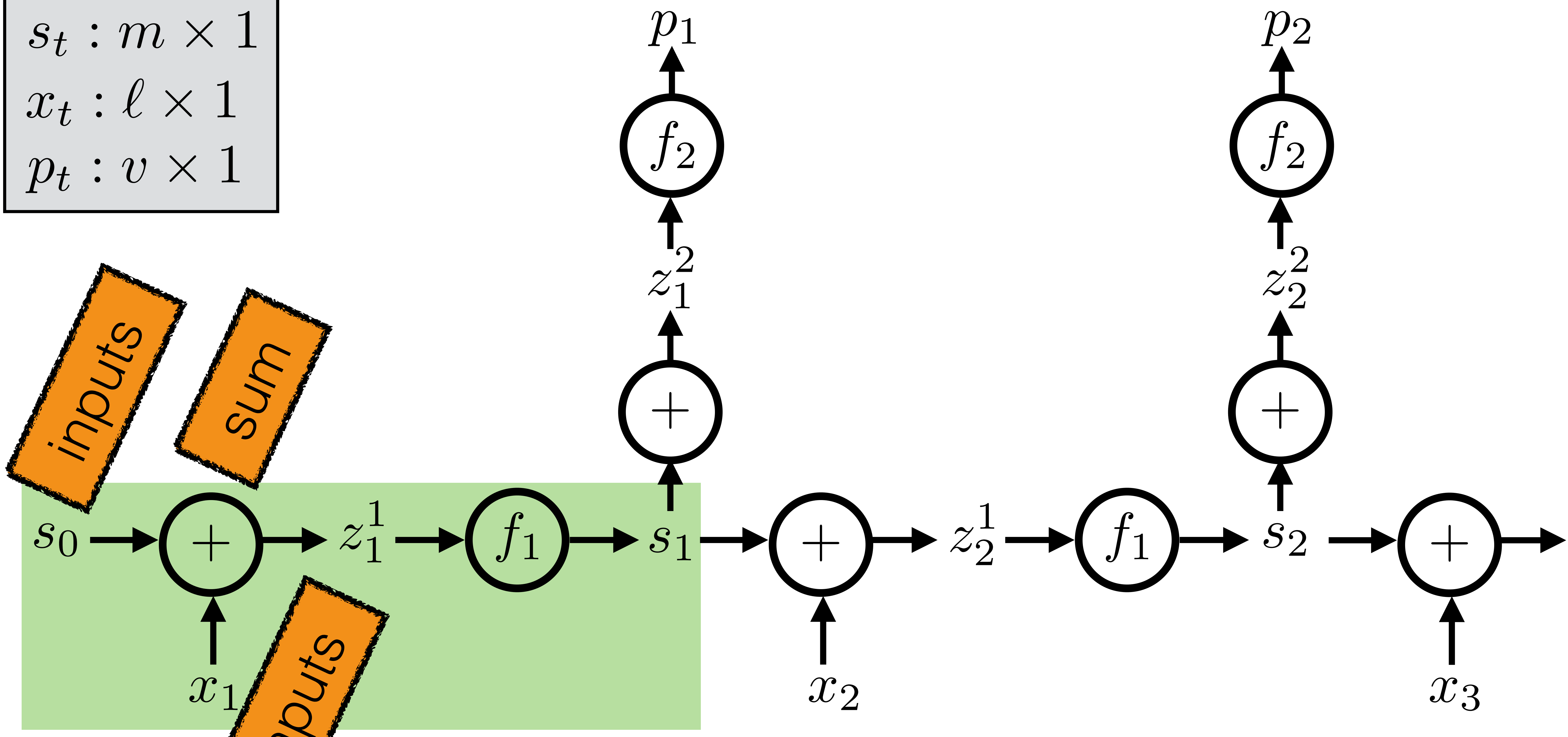
$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2})$$

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



- Example: Alphabet of ℓ chars; state is last c chars ($m = c\ell$)

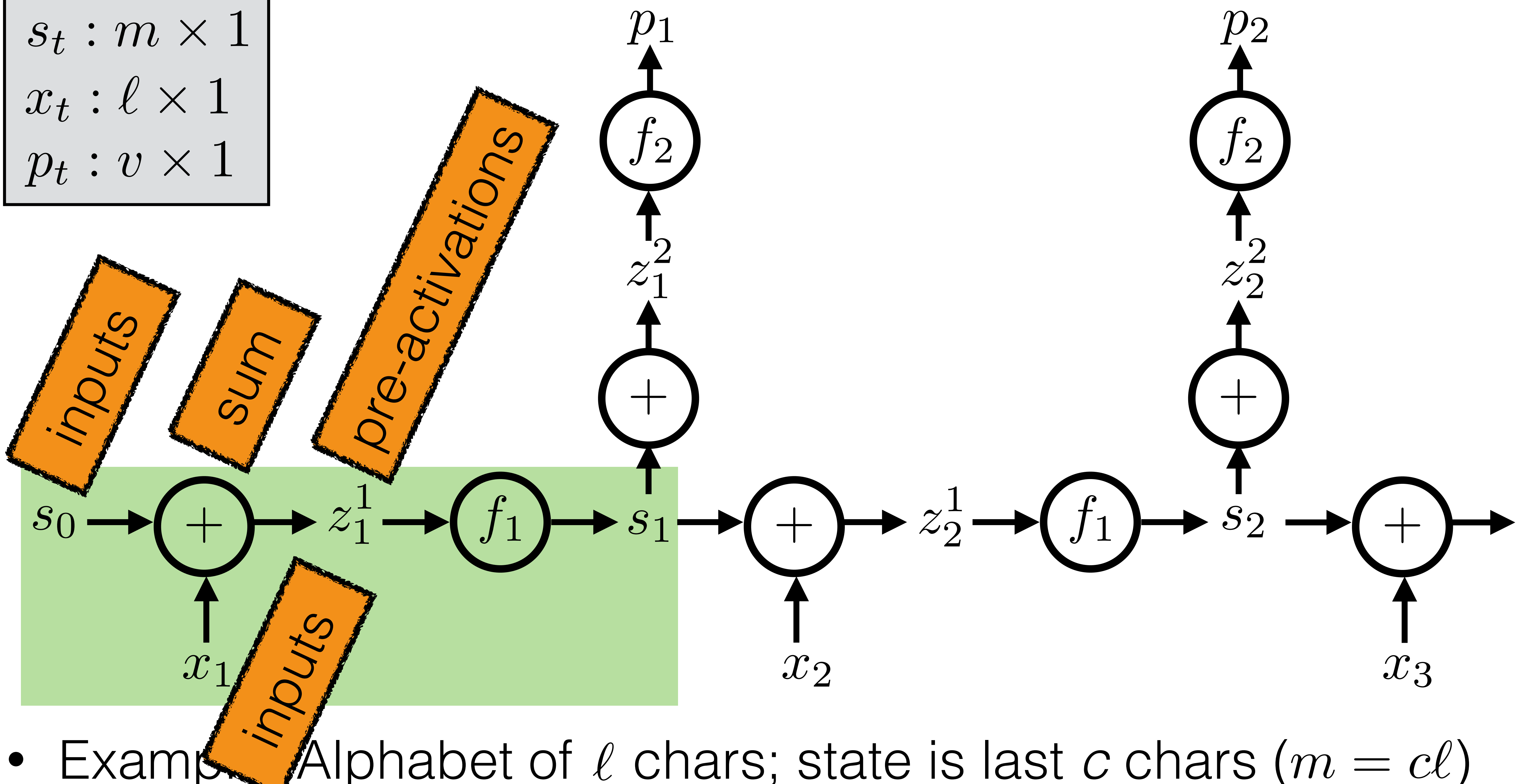
$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2})$$

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



- Example: Alphabet of ℓ chars; state is last c chars ($m = c\ell$)

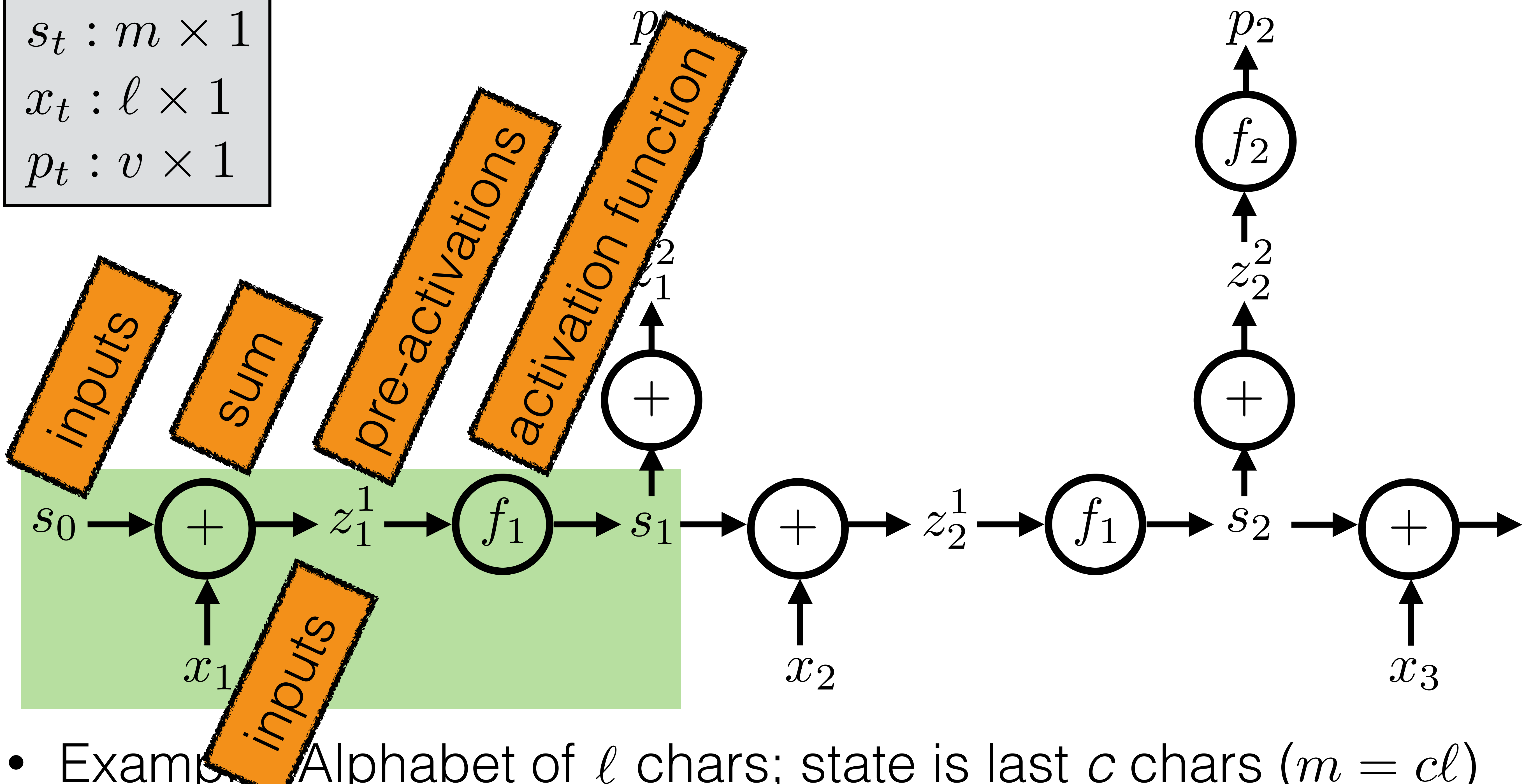
$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2})$$

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



- Examp... Alphabet of ℓ chars; state is last c chars ($m = c\ell$)

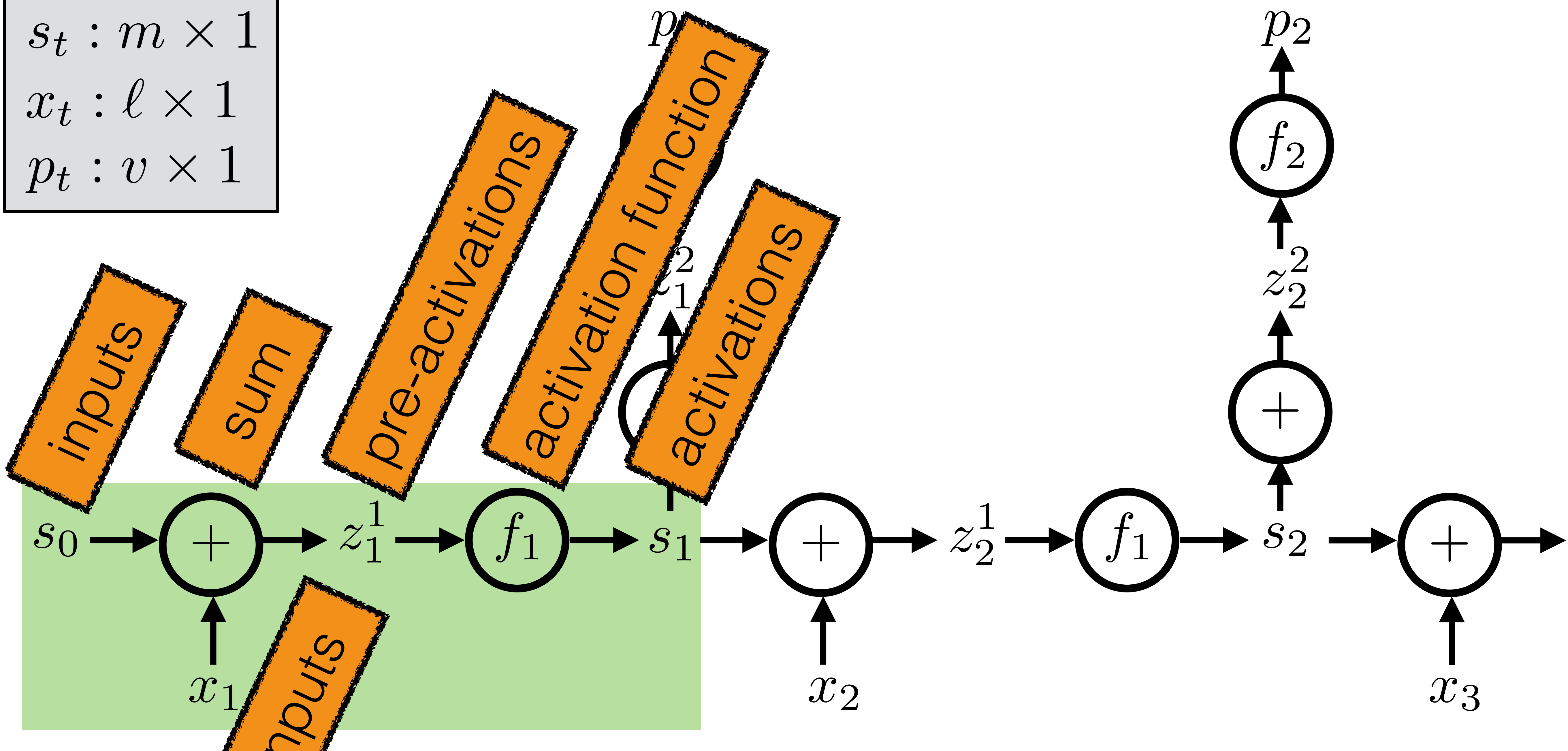
$$s_t = f_1(\underbrace{W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}}_{z_t^1})$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2})$$

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



- Example: Alphabet of ℓ chars; state is last c chars ($m = c\ell$)

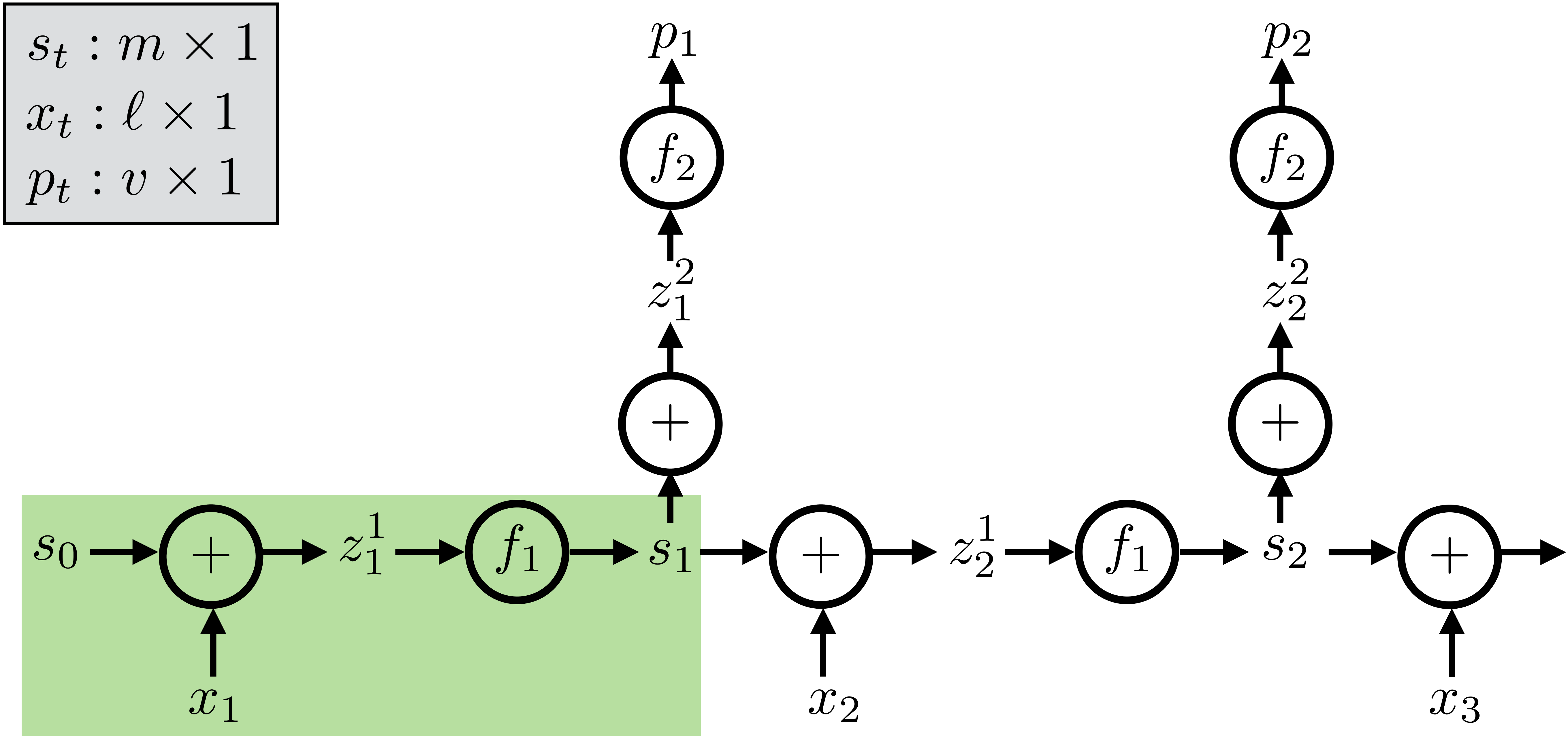
$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2})$$

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$
$x_t : \ell \times 1$
$p_t : v \times 1$



- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2})$$

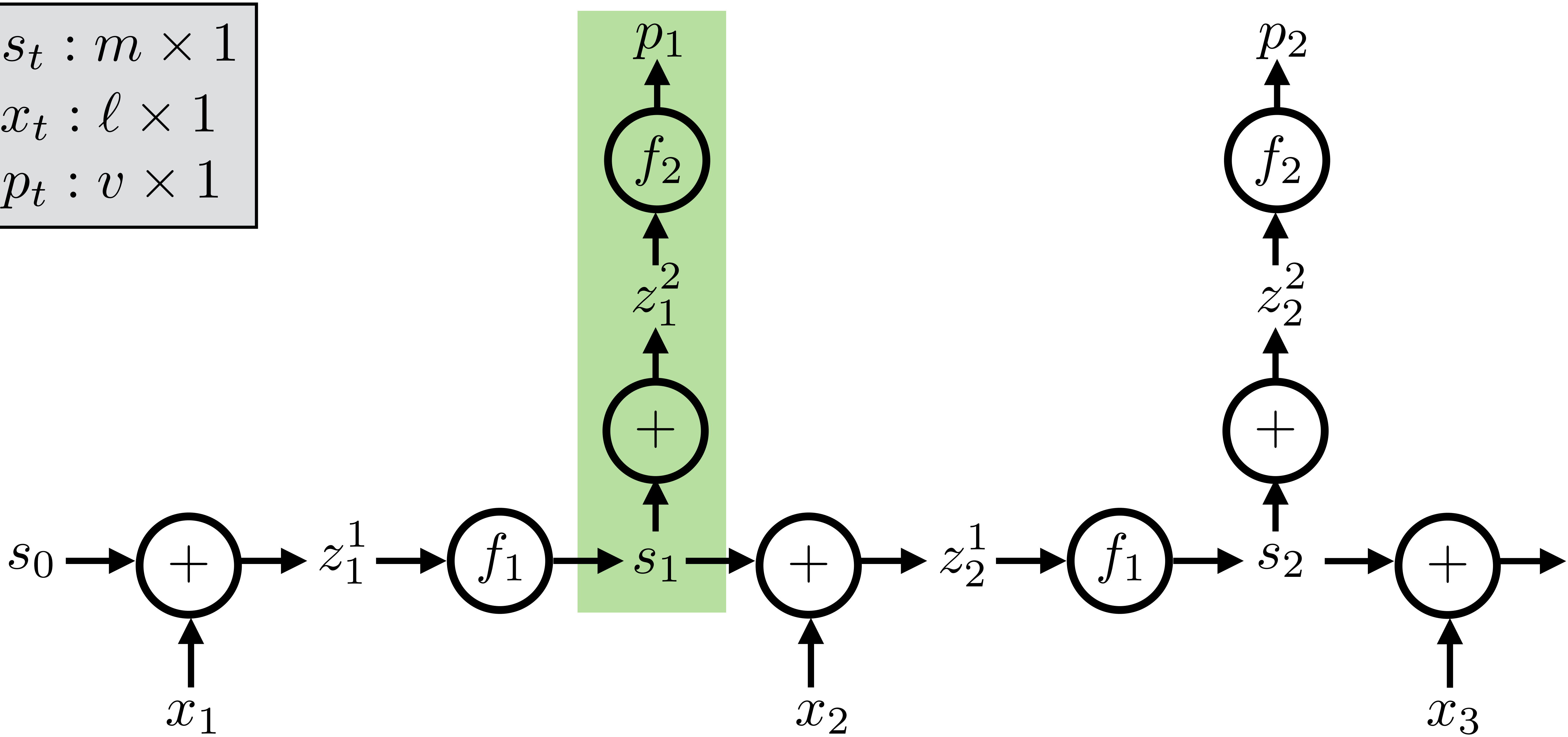
$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

Recurrent neural network

$s_t : m \times 1$

$x_t : \ell \times 1$

$p_t : v \times 1$



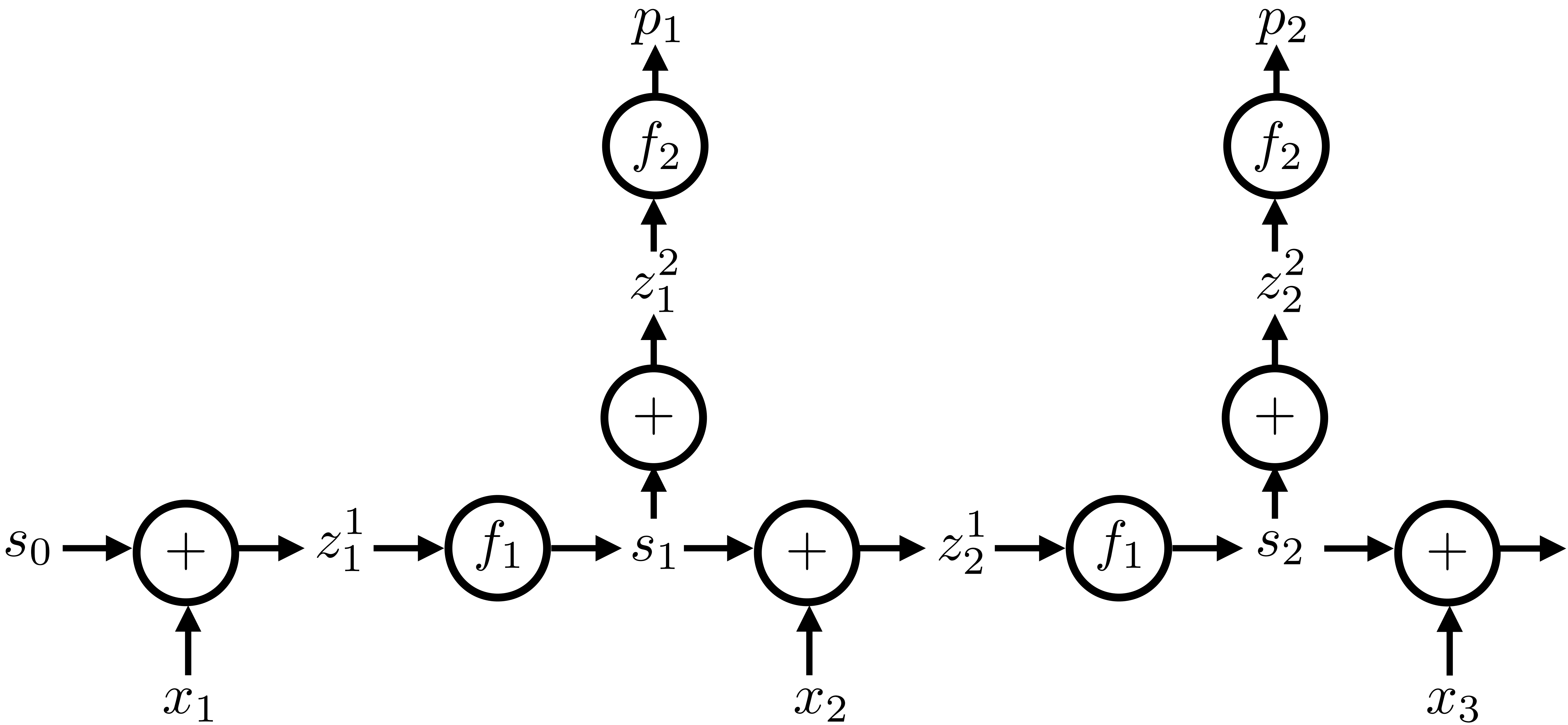
- Example: Alphabet of ℓ chars; state is last *c* chars ($m = c\ell$)

$$s_t = f_1\left(\underbrace{W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}}_{z_t^1}\right)$$

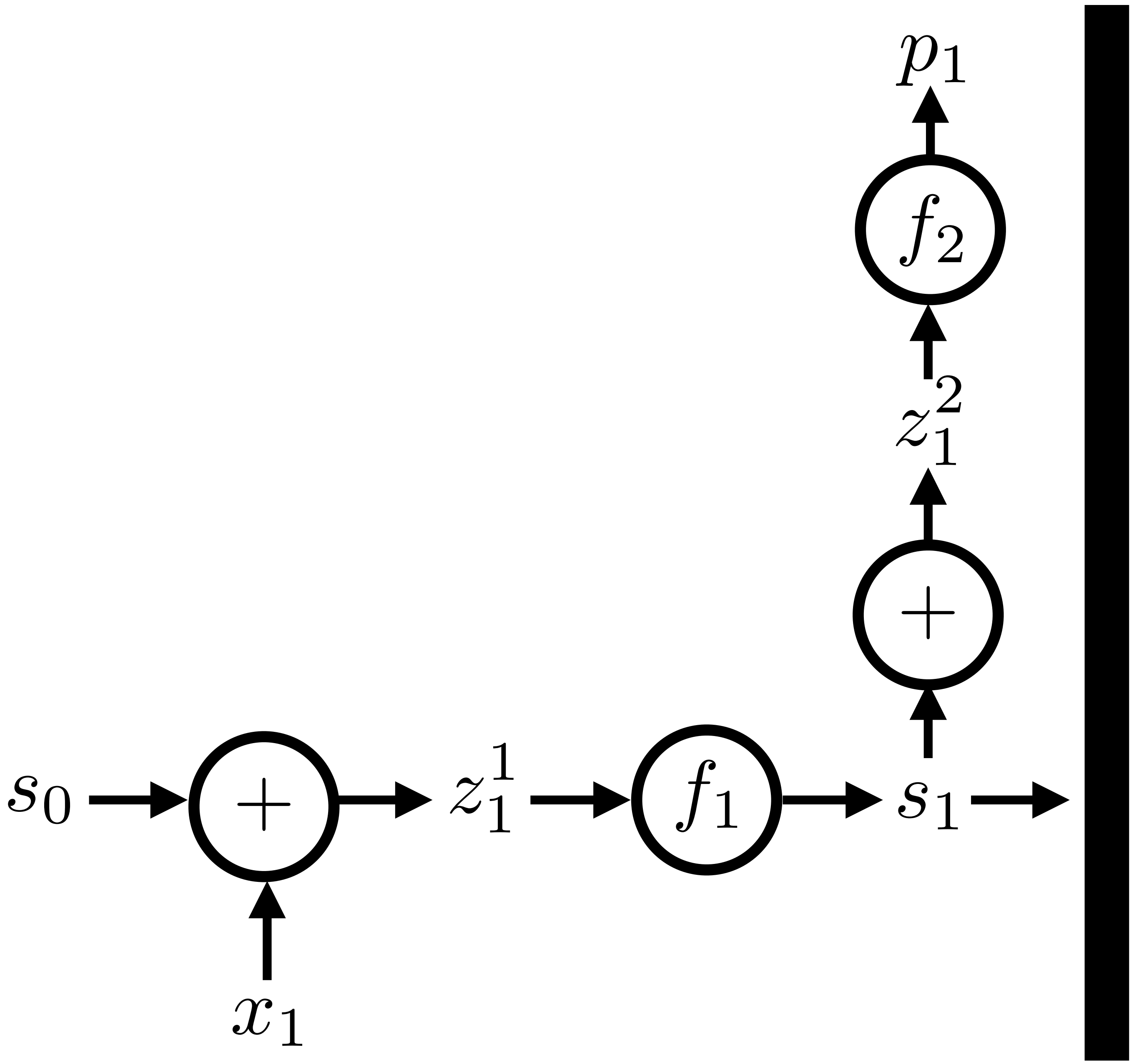
$$p_t = f_2(\underbrace{W^o s_t + W_0^o}_{z_t^2})$$

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

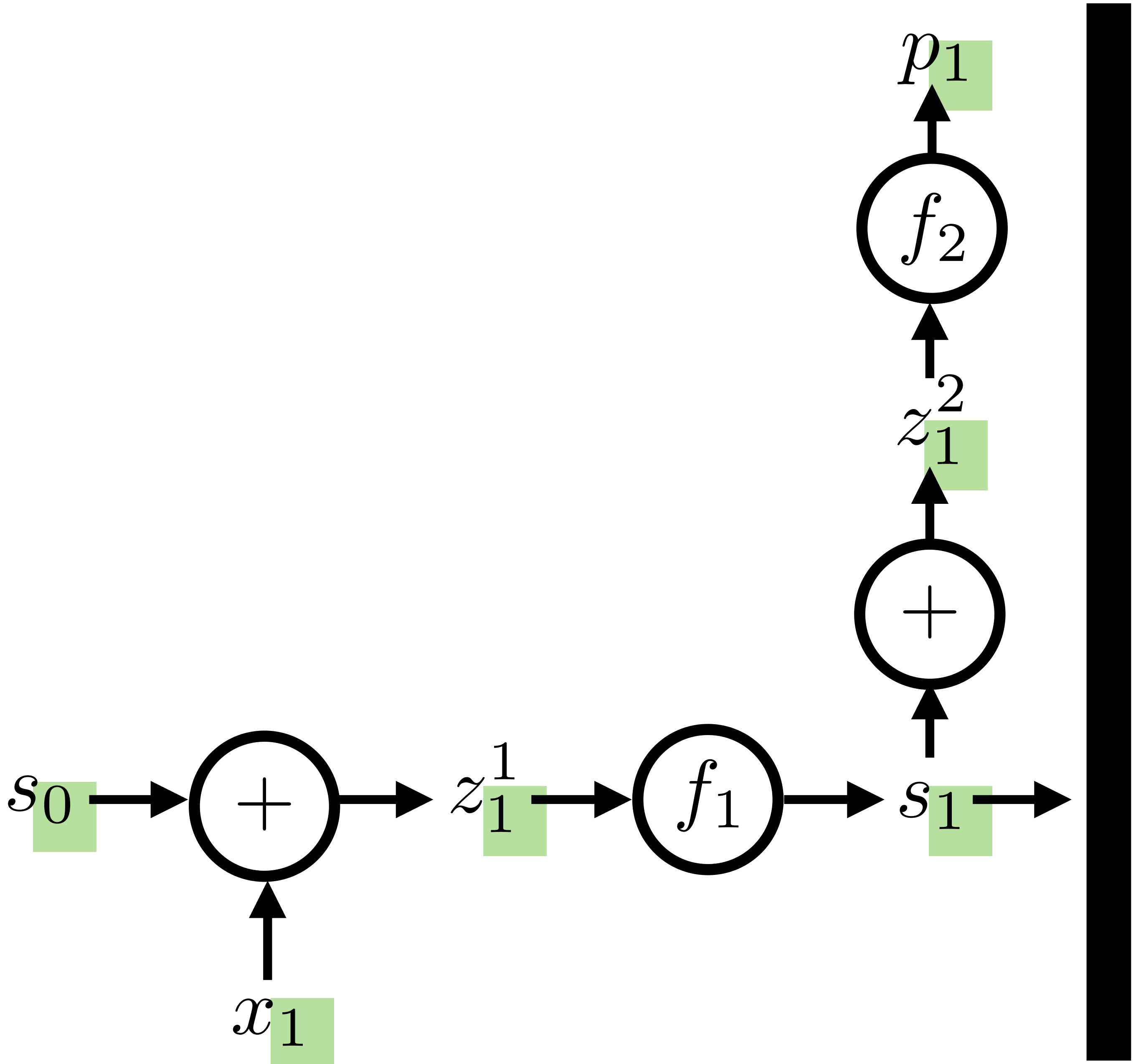
Recurrent neural network



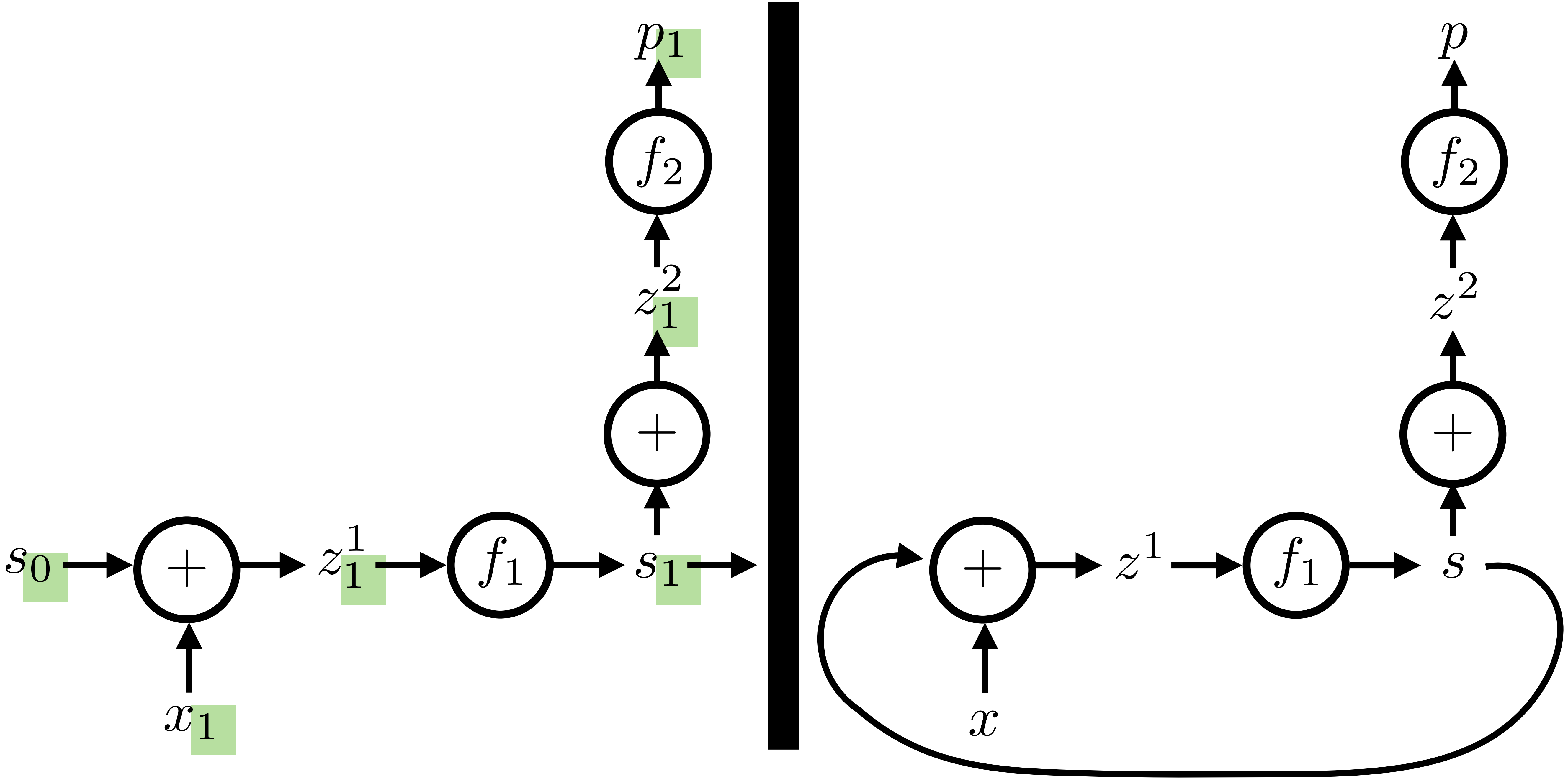
Recurrent neural network



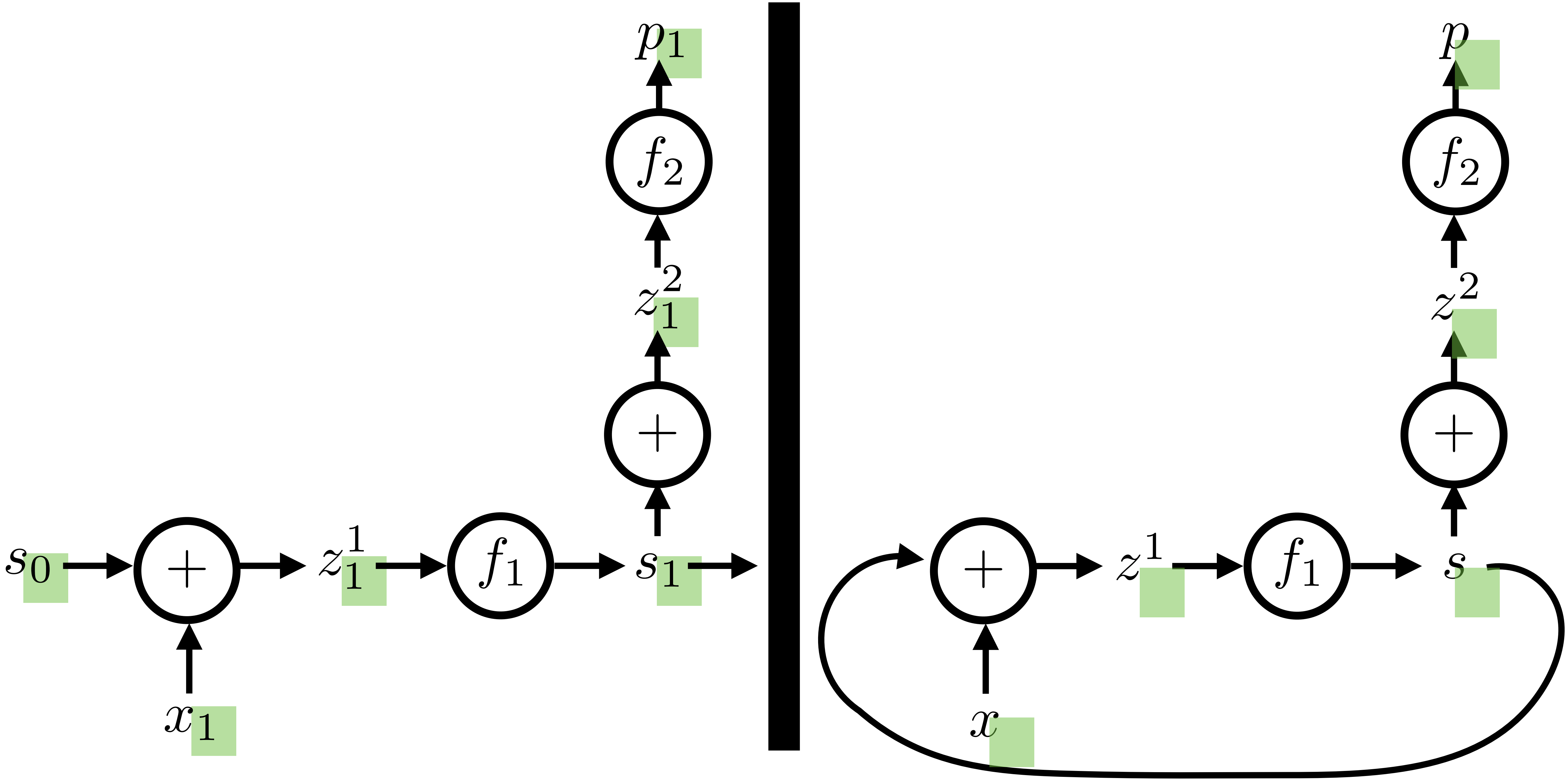
Recurrent neural network



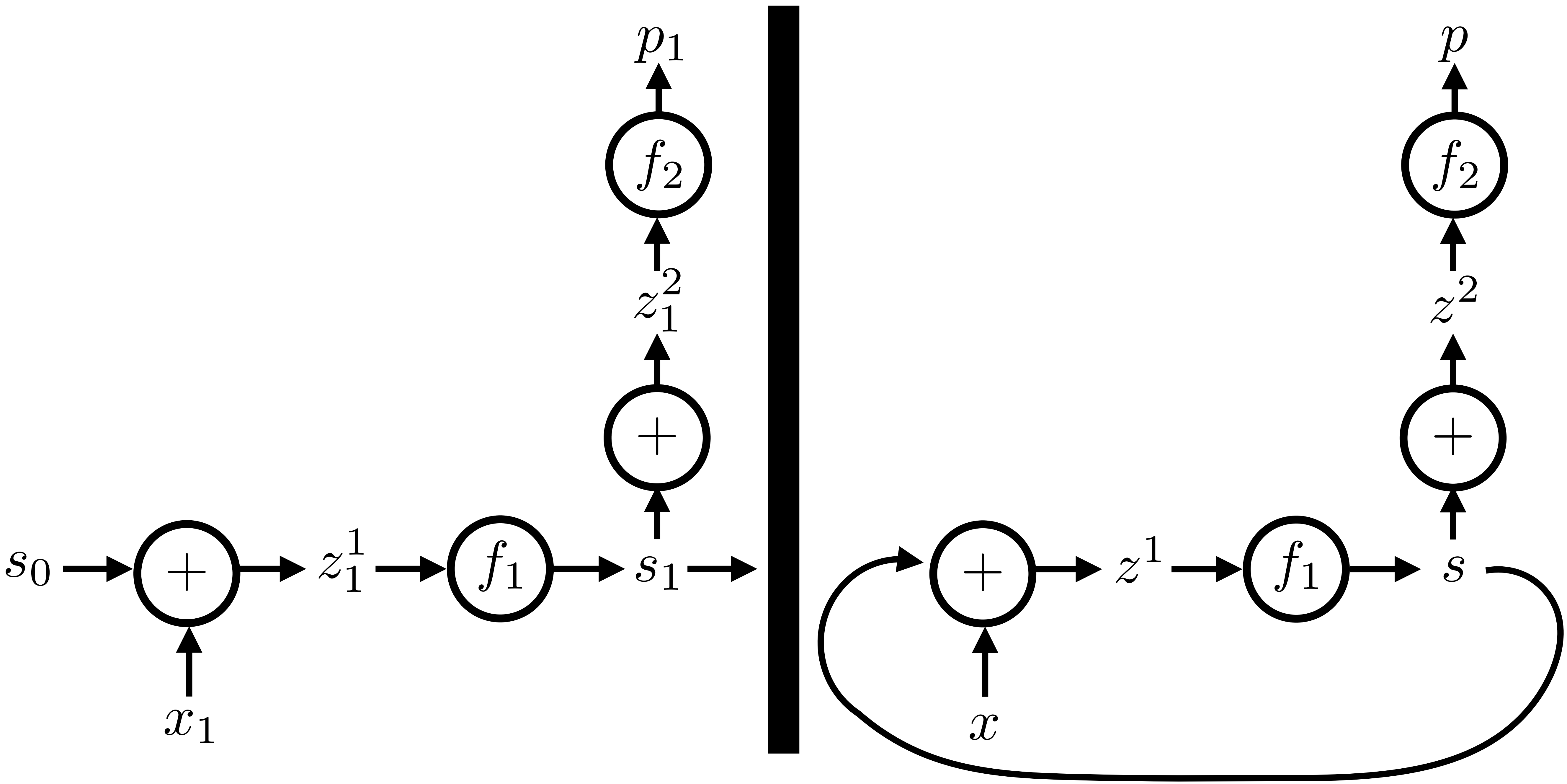
Recurrent neural network



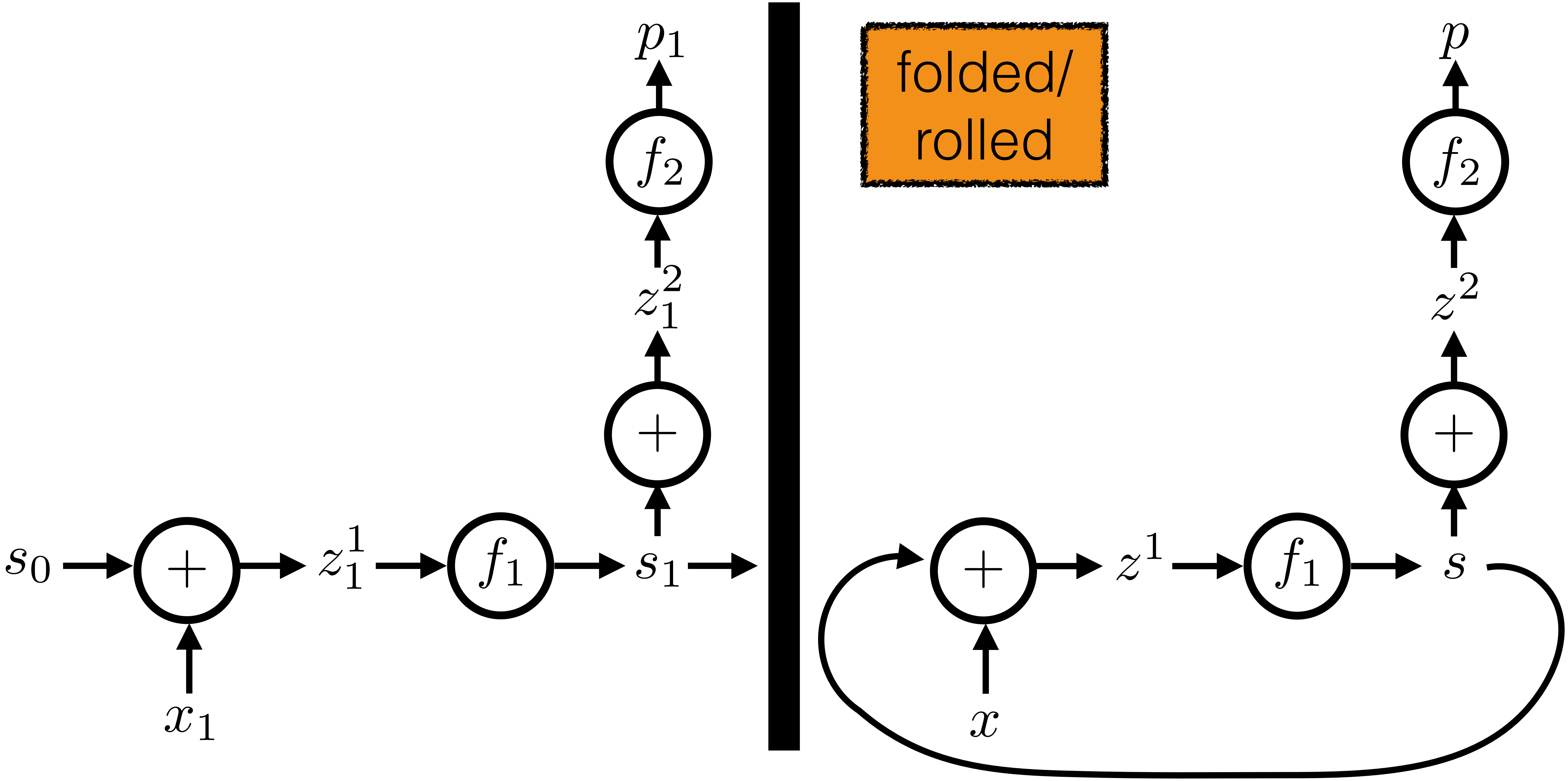
Recurrent neural network



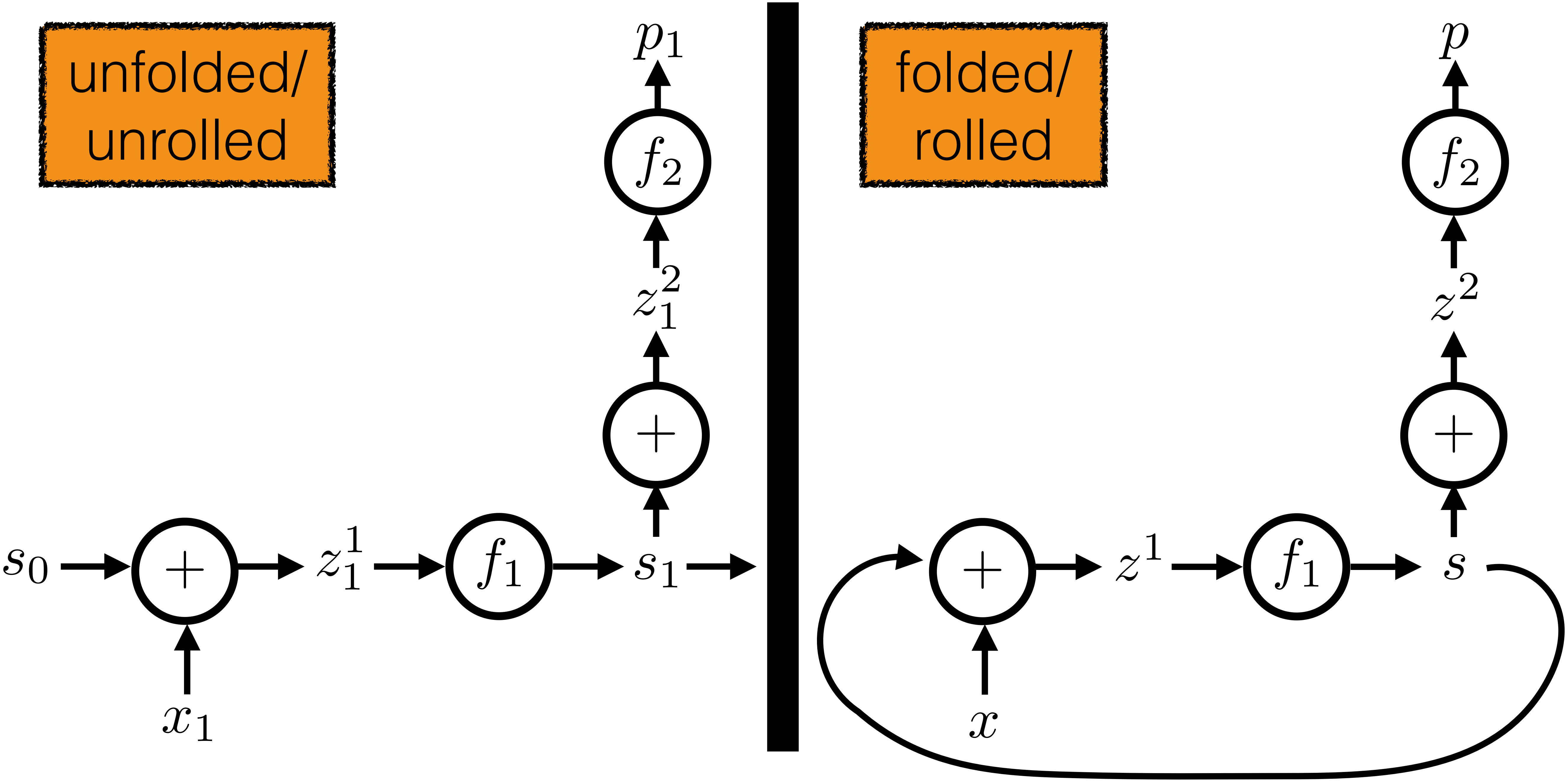
Recurrent neural network



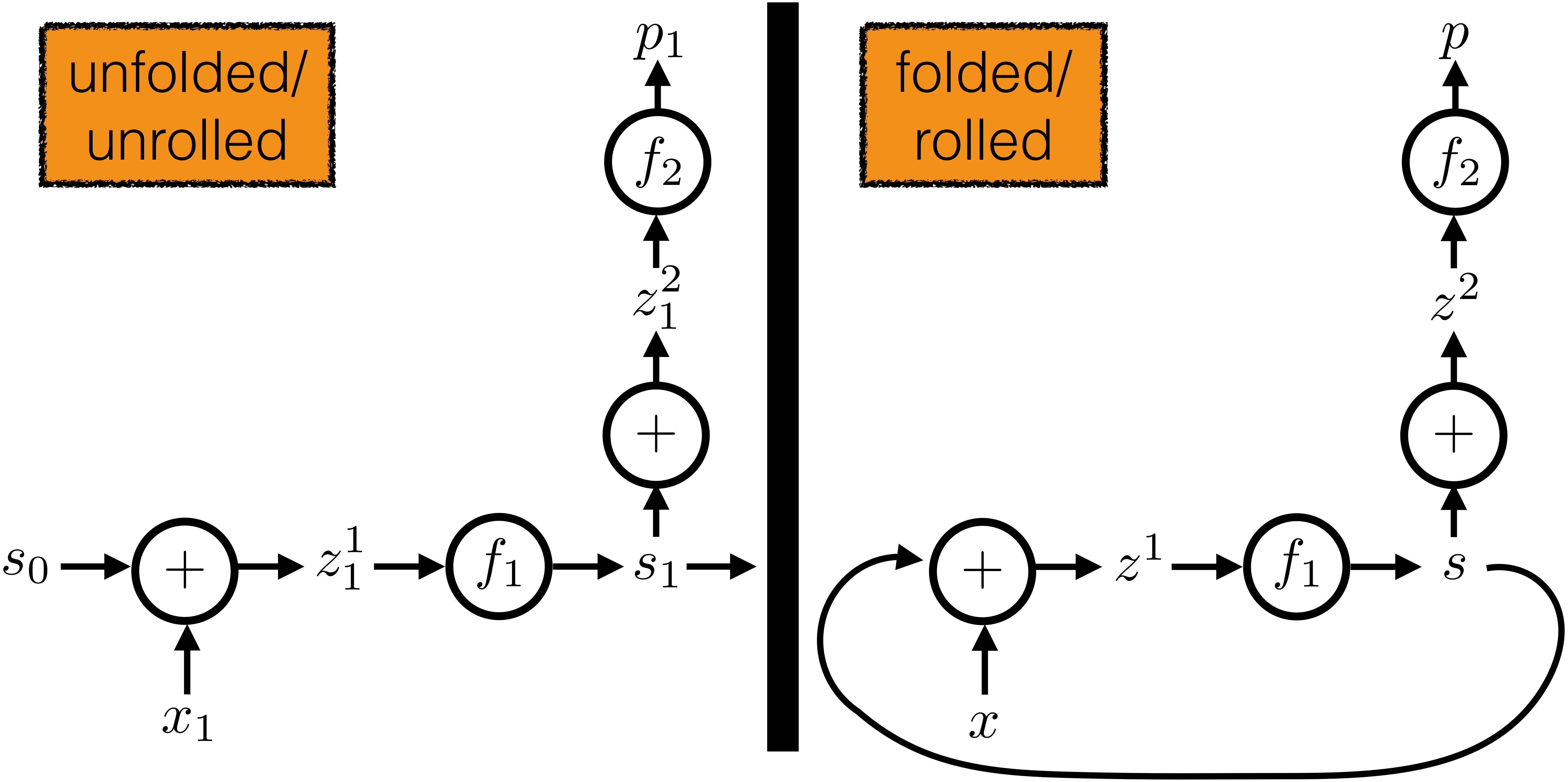
Recurrent neural network



Recurrent neural network

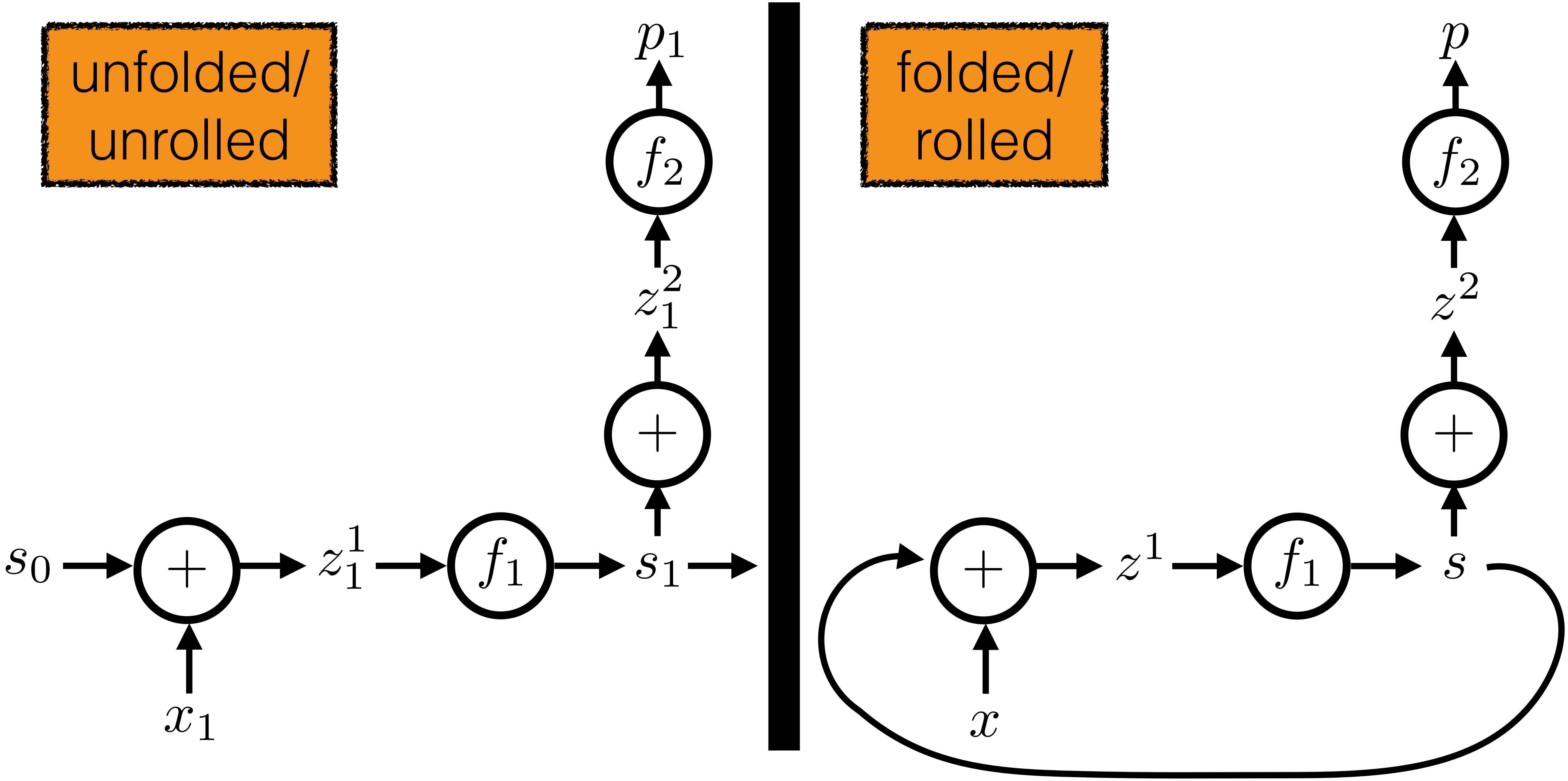


Recurrent neural network



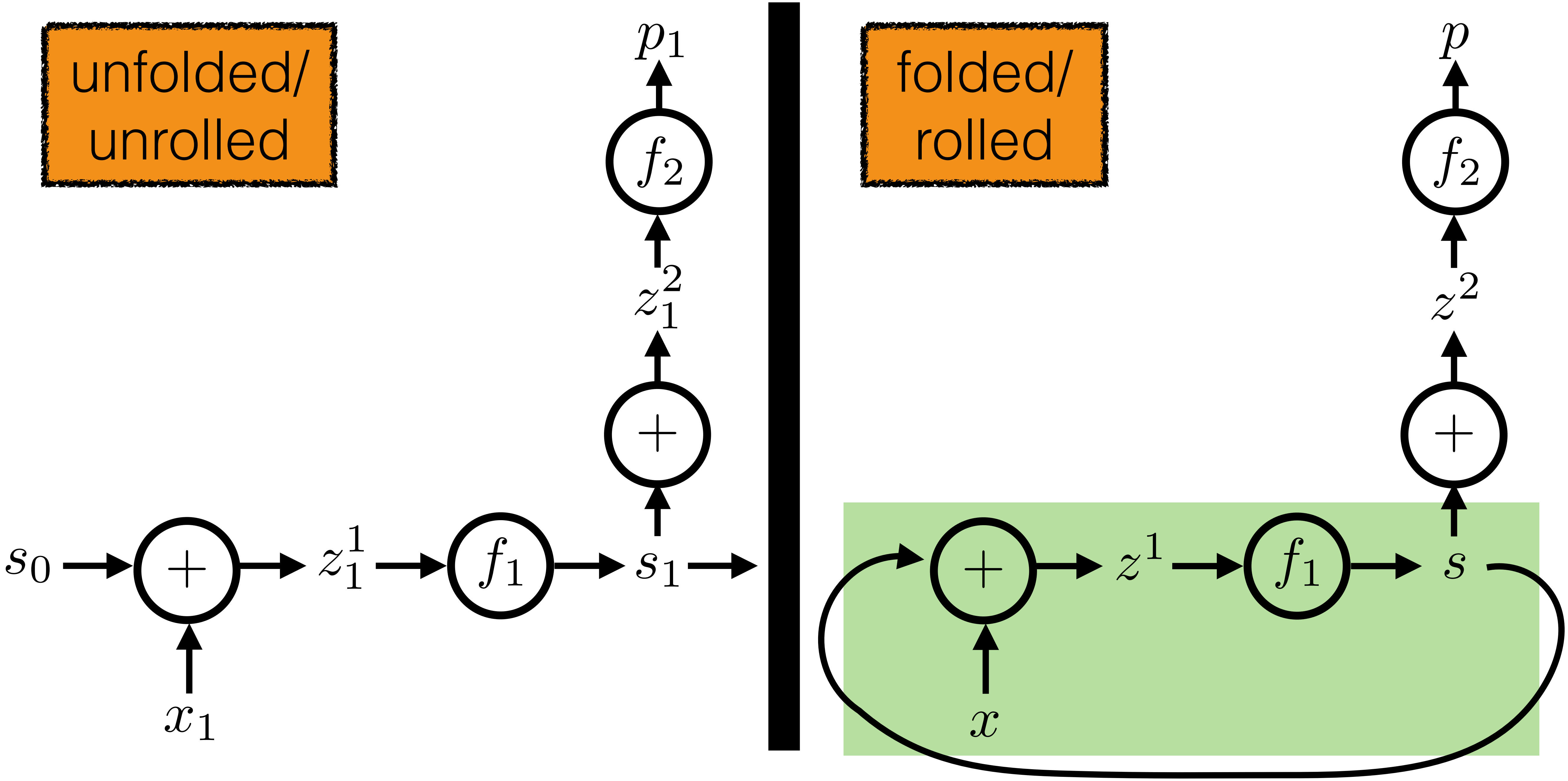
- Compare to:

Recurrent neural network



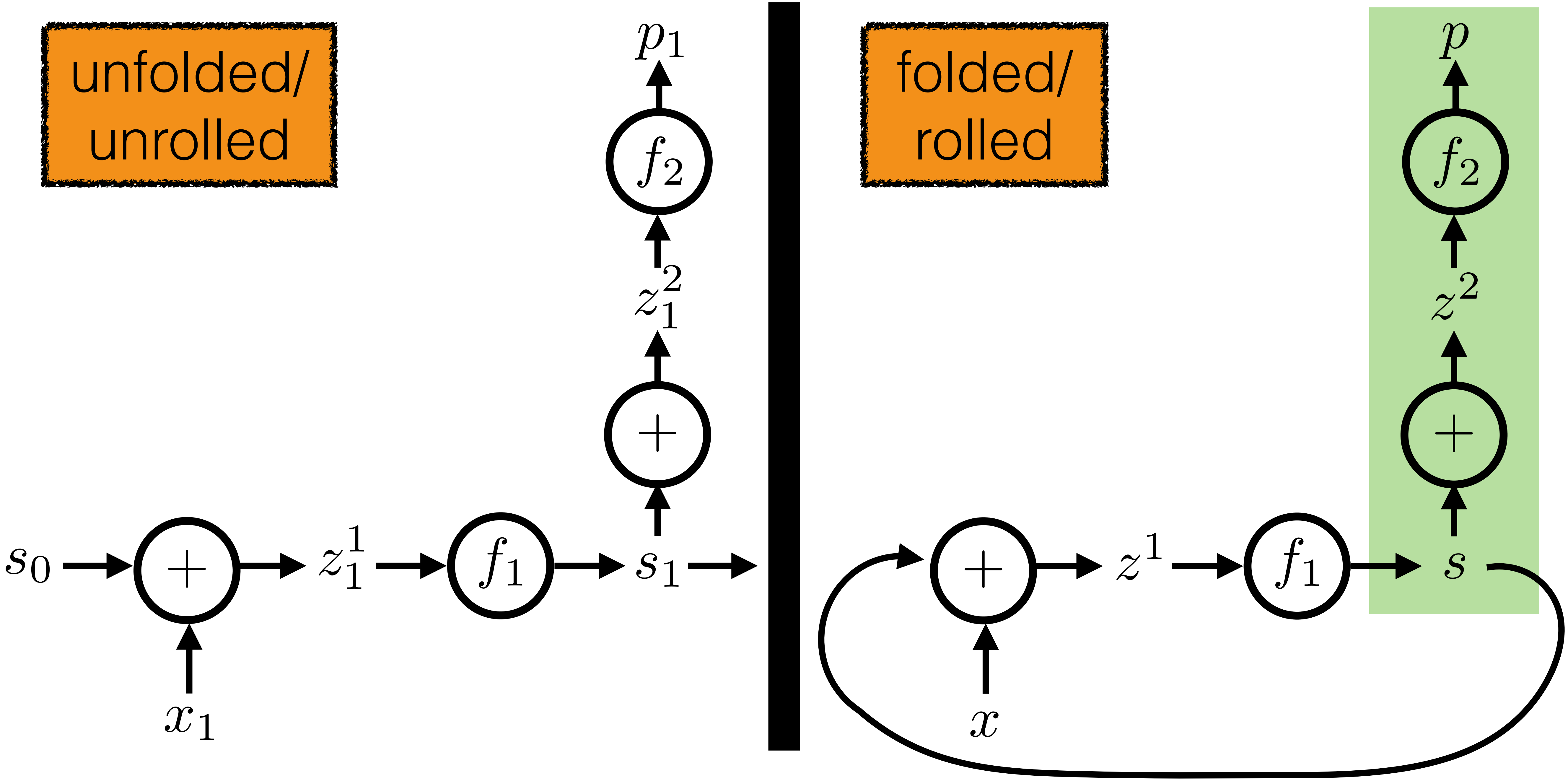
- Compare to:
  - Feedforward neural networks

Recurrent neural network



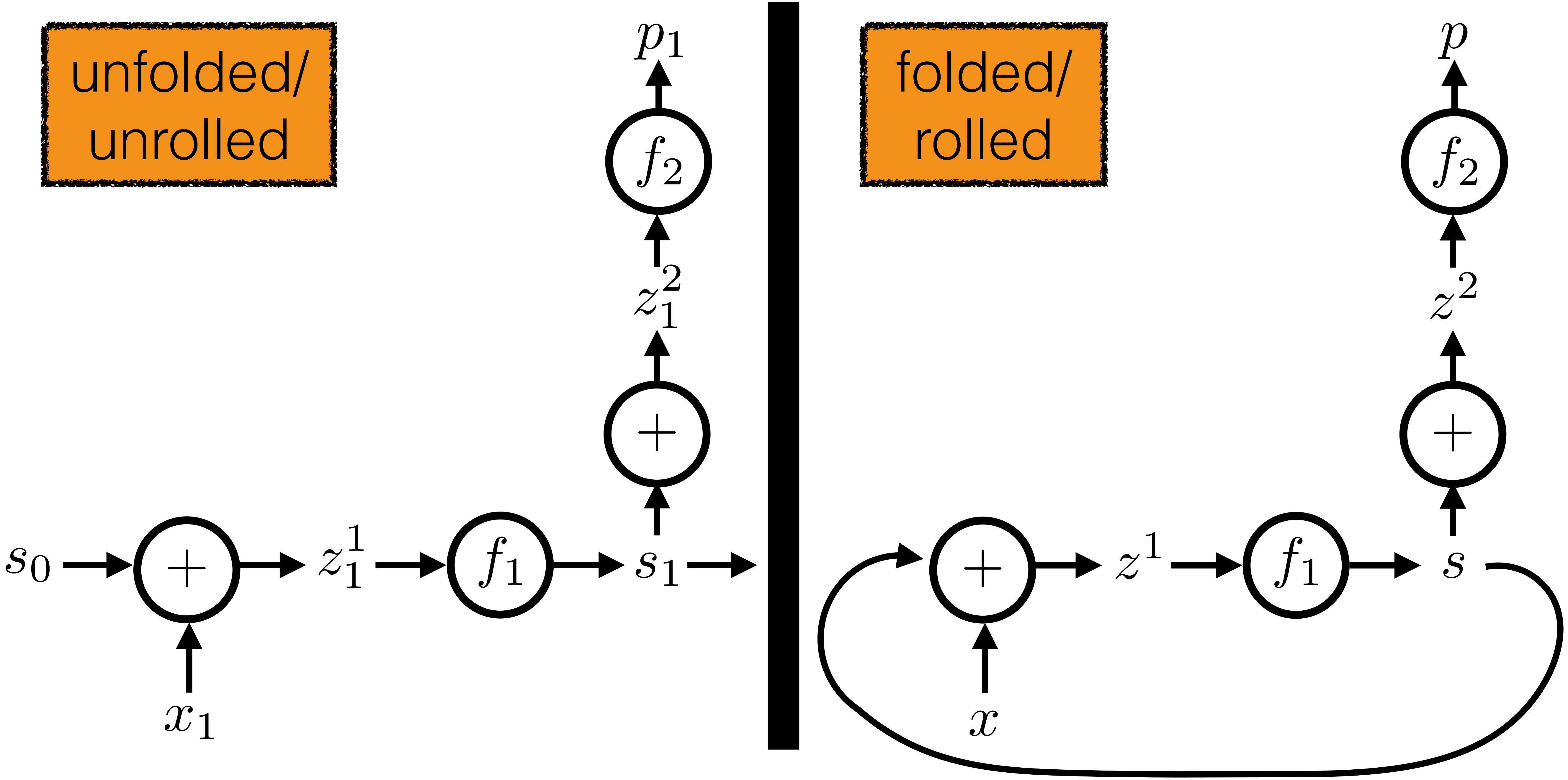
- Compare to:
    - Feedforward neural networks

Recurrent neural network



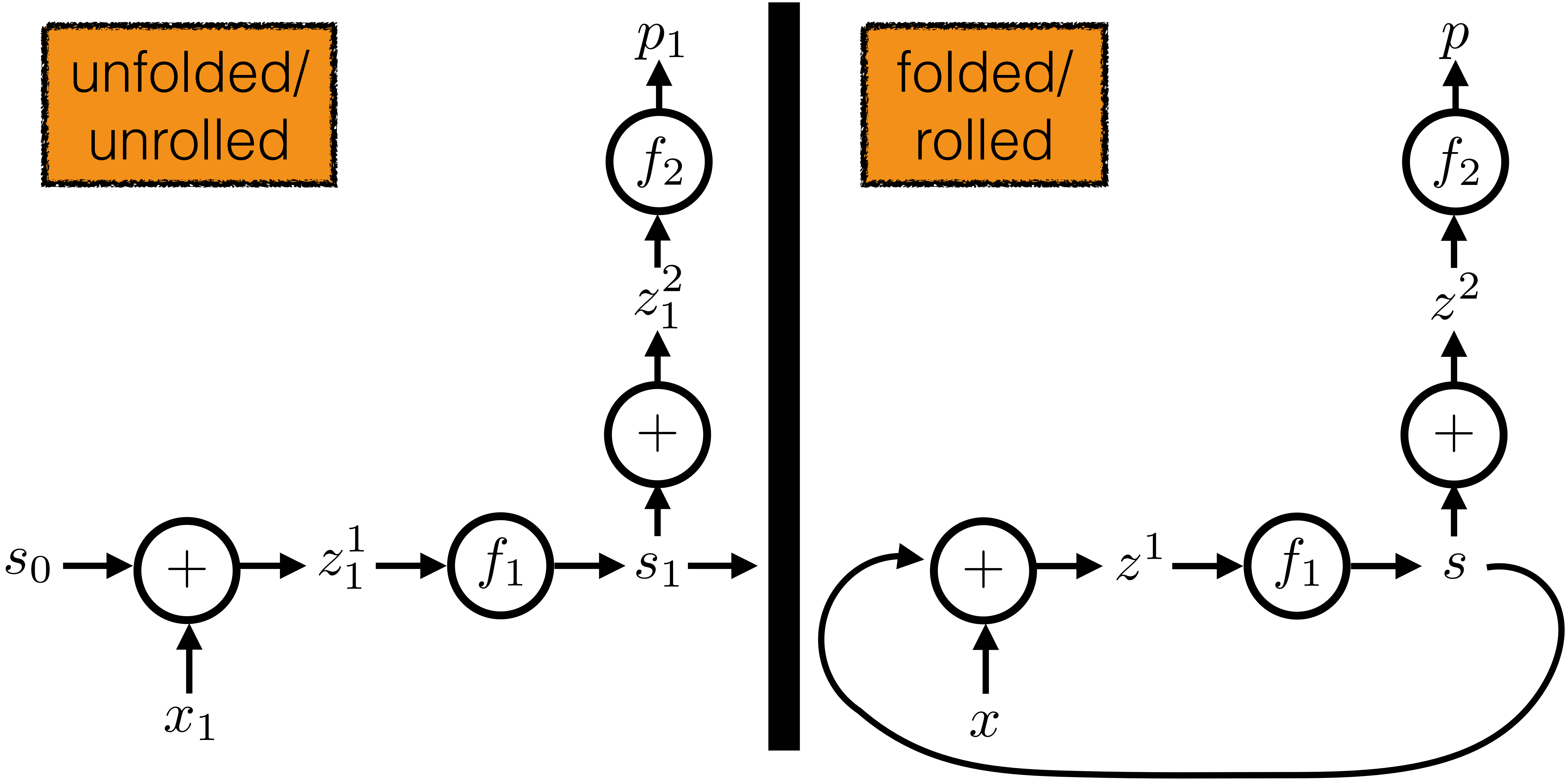
- Compare to:
  - Feedforward neural networks

Recurrent neural network

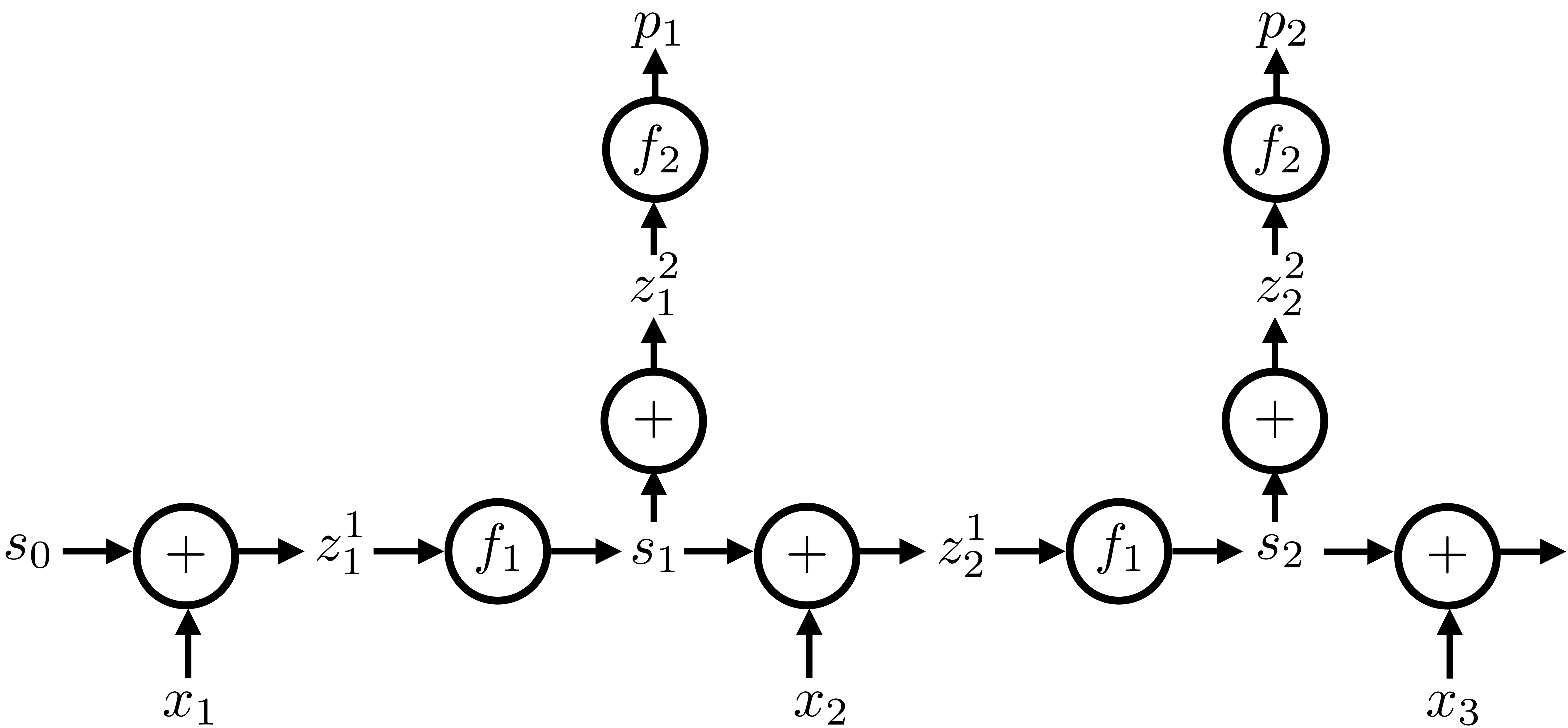


- Compare to:
    - Feedforward neural networks
    - Convolutional neural networks

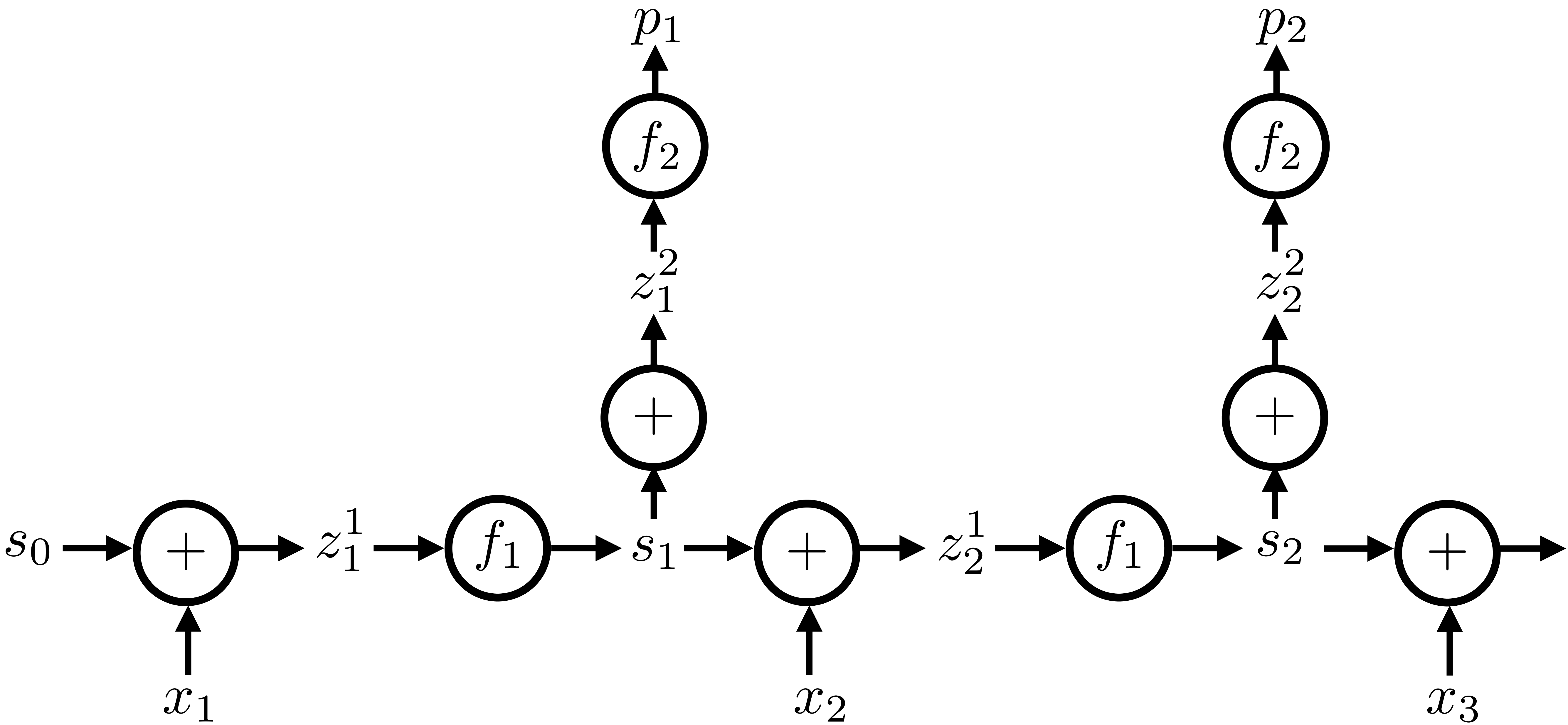
Recurrent neural network



- Compare to:
  - Feedforward neural networks
  - Convolutional neural networks
  - Reinforcement learning

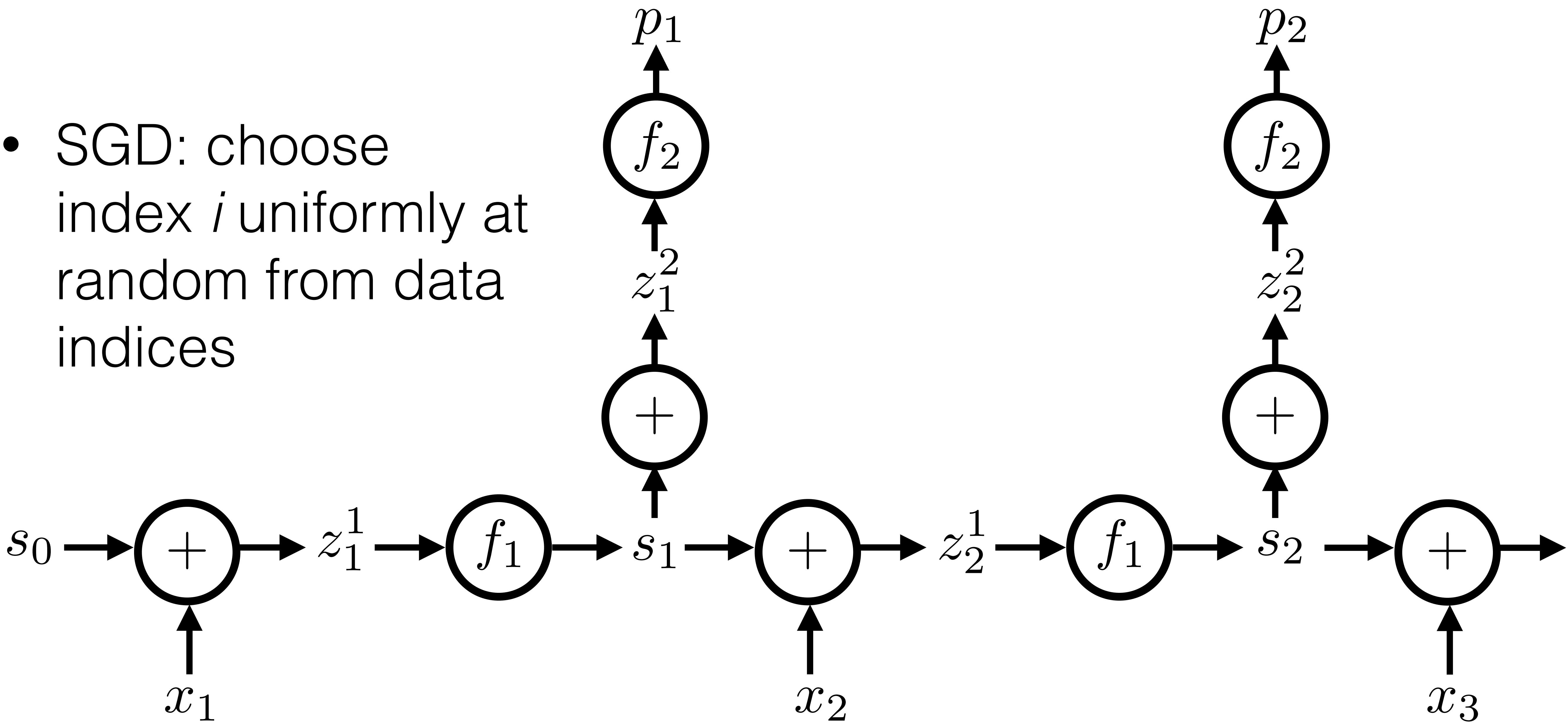

p_1
p_2
f_2
f_2
z_1^2
z_2^2
$+$
$+$
s_0
$+$
z_1^1
f_1
s_1
$+$
z_2^1
f_1
s_2
$+$
x_1
x_2
x_3

RNNs: a taste of backpropagation



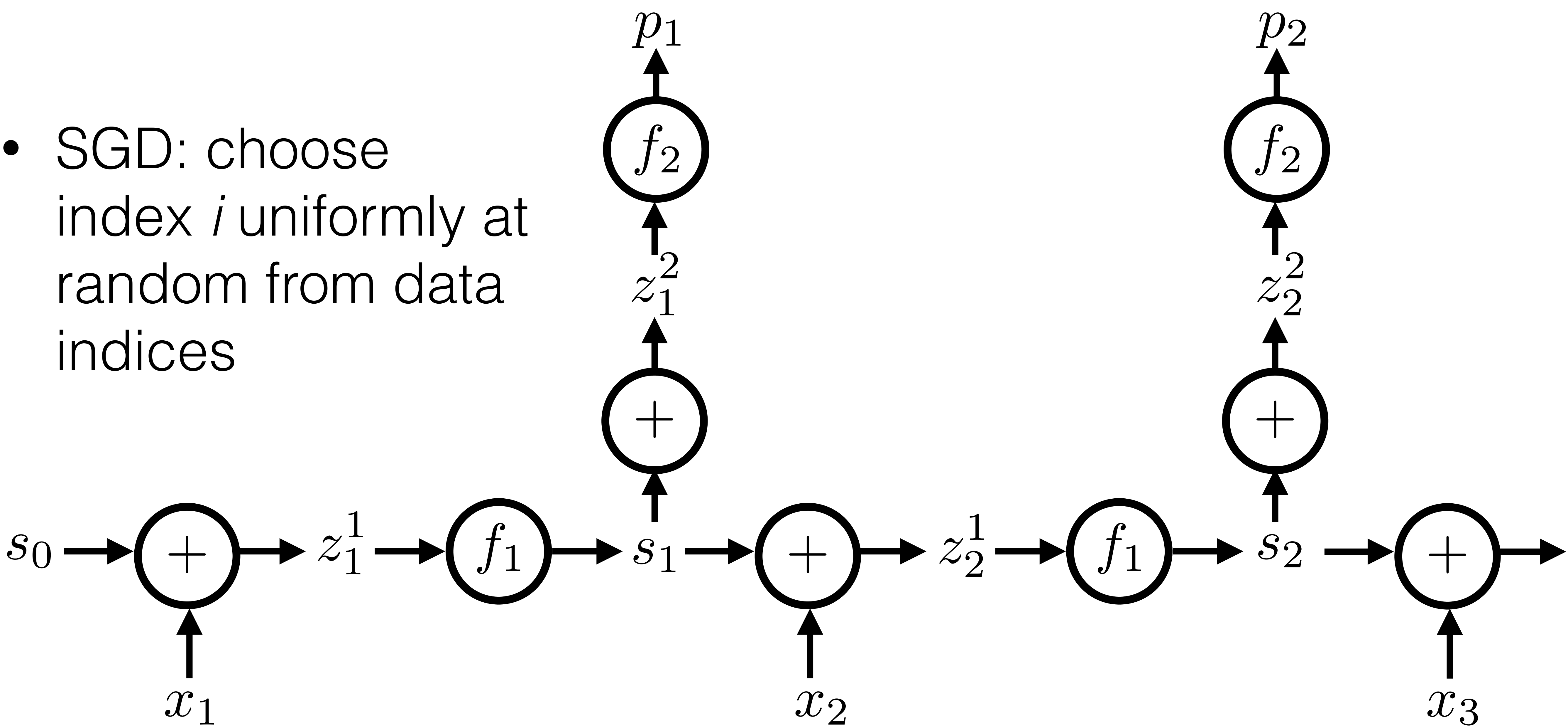
RNNs: a taste of backpropagation

- SGD: choose index *i* uniformly at random from data indices



RNNs: a taste of backpropagation

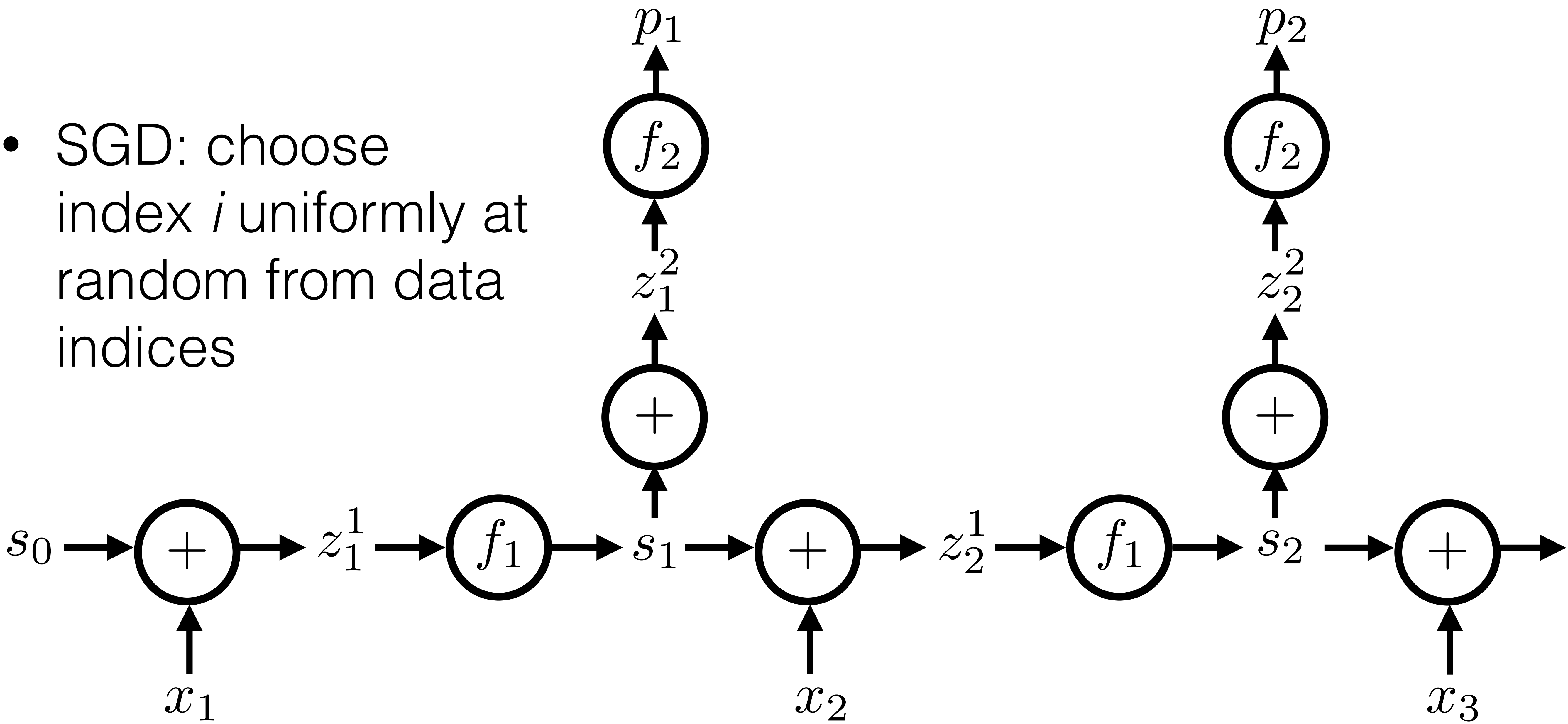
- SGD: choose index *i* uniformly at random from data indices



$L_{\text{seq}}(p^{(i)}, y^{(i)})$

RNNs: a taste of backpropagation

- SGD: choose index *i* uniformly at random from data indices

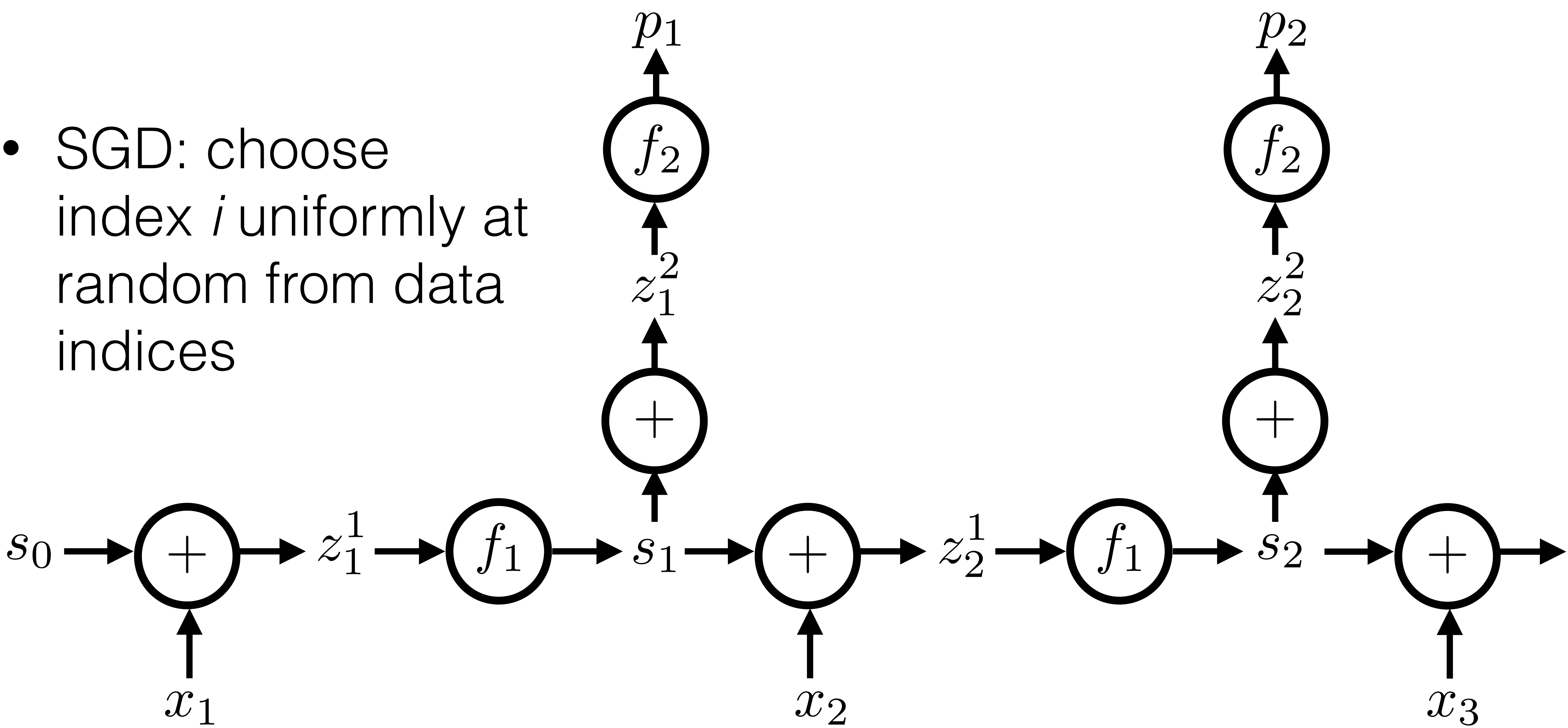


$$L_{\text{seq}}(p^{(i)}, y^{(i)})$$

$$\frac{dL_{\text{seq}}(p^{(i)}, y^{(i)})}{dW^{sx}}$$

RNNs: a taste of backpropagation

- SGD: choose index *i* uniformly at random from data indices

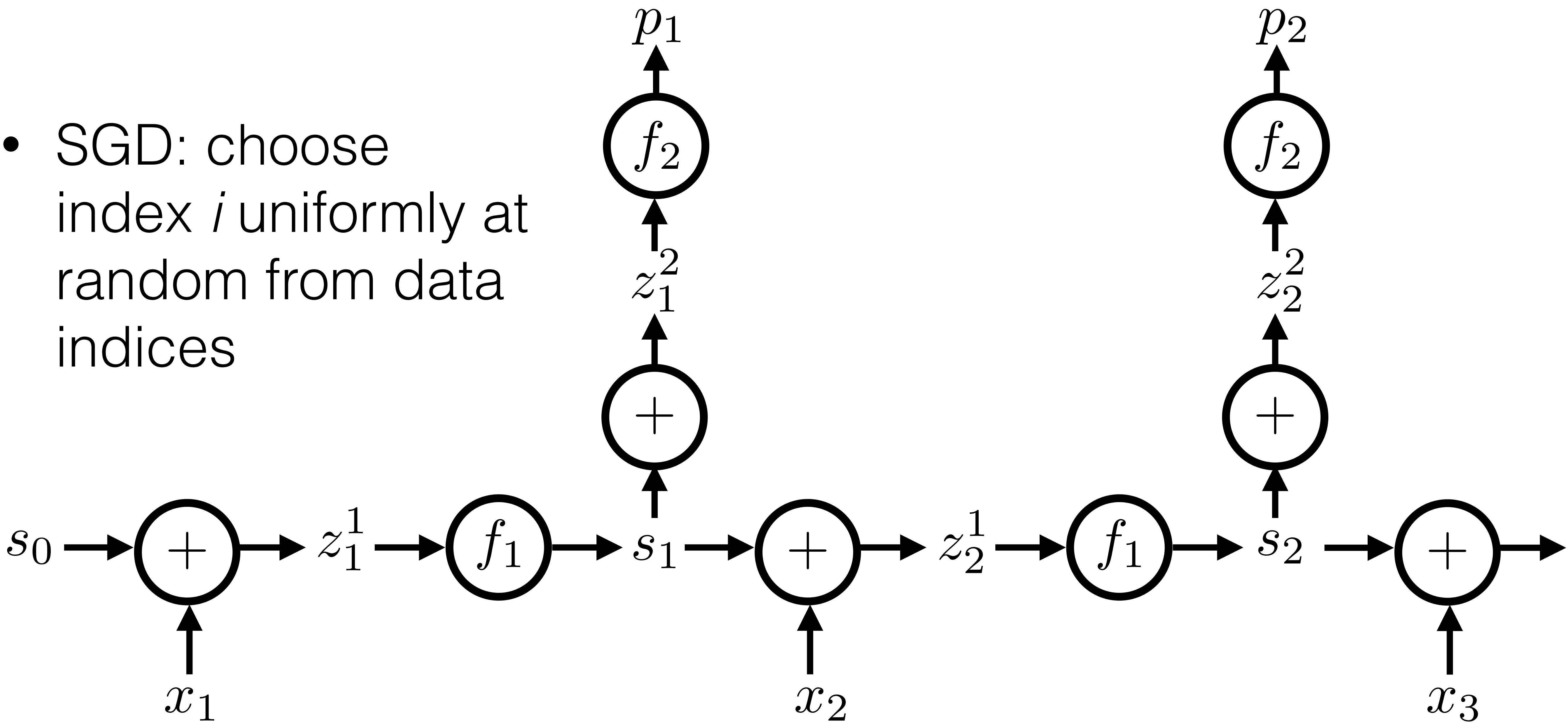


$$L_{\text{seq}}(p^{(i)}, y^{(i)})$$

$$\frac{dL_{\text{seq}}(p^{(i)}, y^{(i)})}{dW^{sx}}$$

RNNs: a taste of backpropagation

- SGD: choose index *i* uniformly at random from data indices

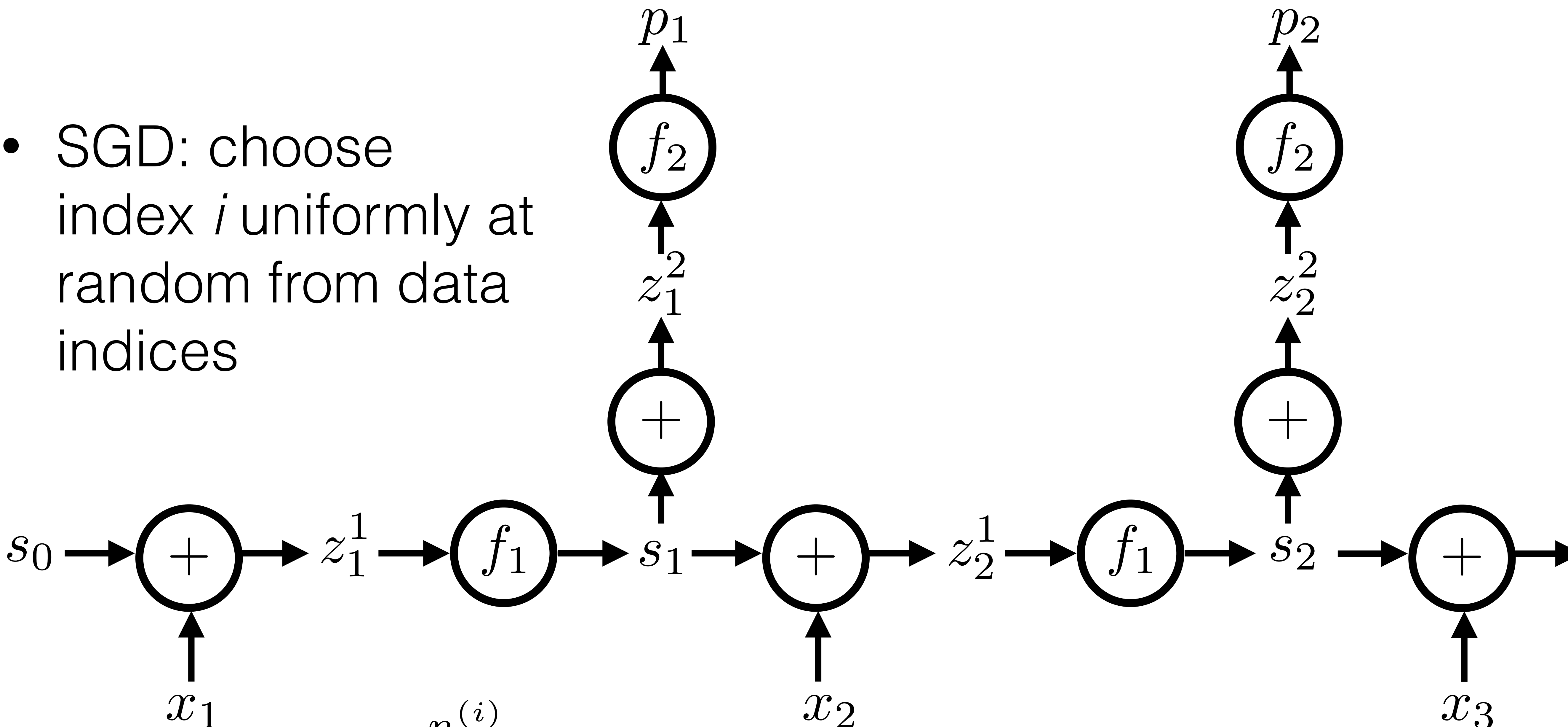


$$L_{\text{seq}}(p^{(i)}, y^{(i)})$$

$$\frac{dL_{\text{seq}}(p^{(i)}, y^{(i)})}{dW^{sx}}$$

RNNs: a taste of backpropagation

- SGD: choose index *i* uniformly at random from data indices

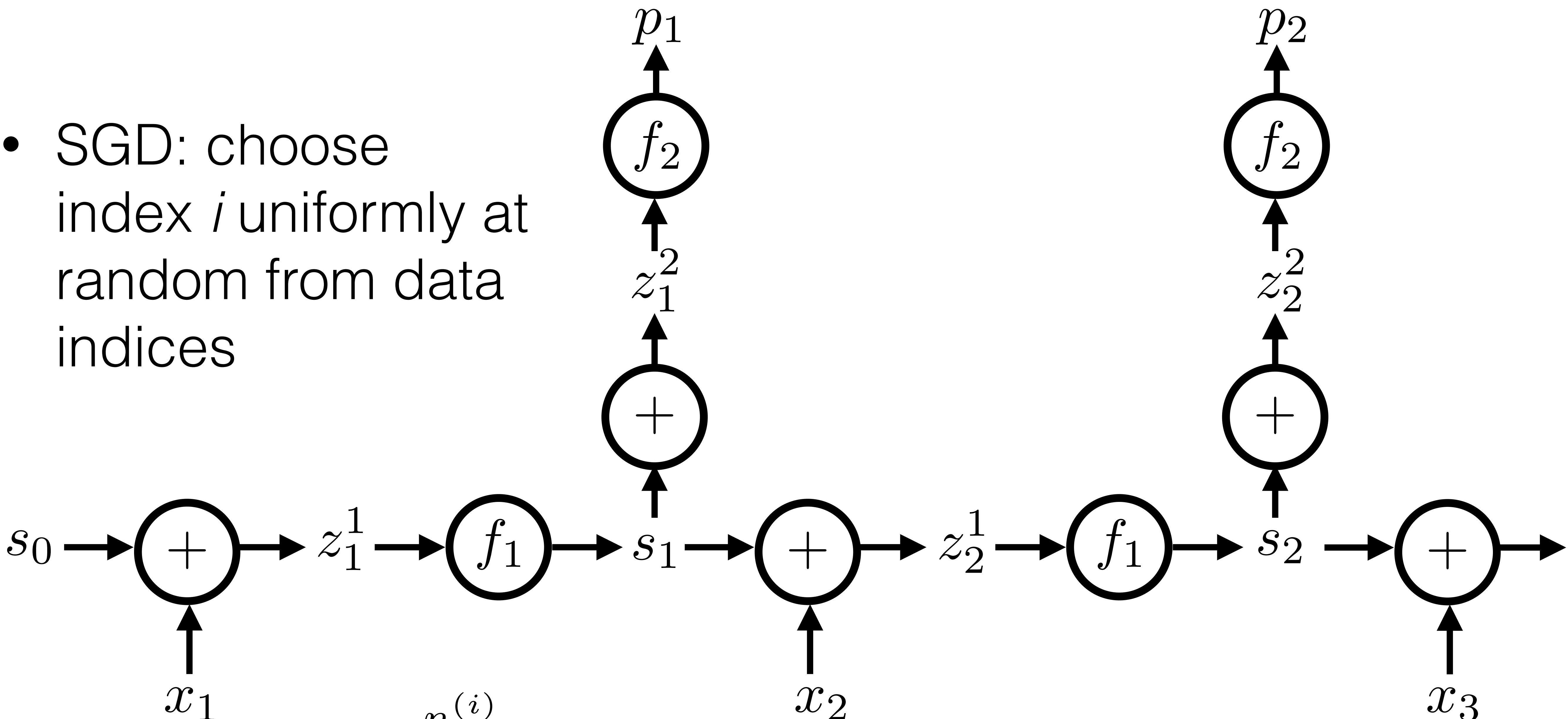


$$L_{\text{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$

$$\frac{dL_{\text{seq}}(p^{(i)}, y^{(i)})}{dW^{sx}}$$

RNNs: a taste of backpropagation

- SGD: choose index *i* uniformly at random from data indices

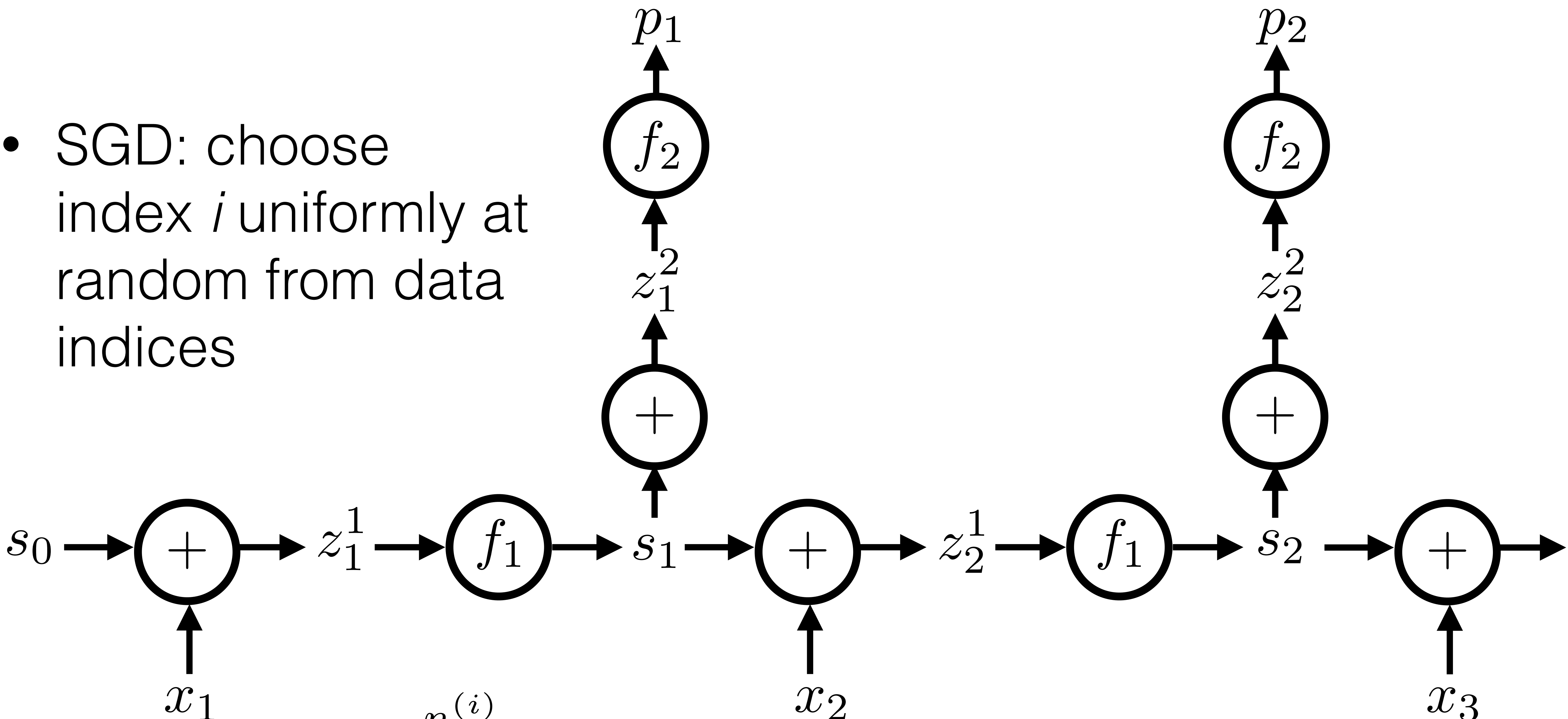


$$L_{\text{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$

$$\frac{dL_{\text{seq}}(p^{(i)}, y^{(i)})}{dW^{sx}} = \sum_{t=1}^{n^{(i)}} \frac{dL_{\text{elt}}(p_t^{(i)}, y_t^{(i)})}{dW^{sx}}$$

RNNs: a taste of backpropagation

- SGD: choose index *i* uniformly at random from data indices



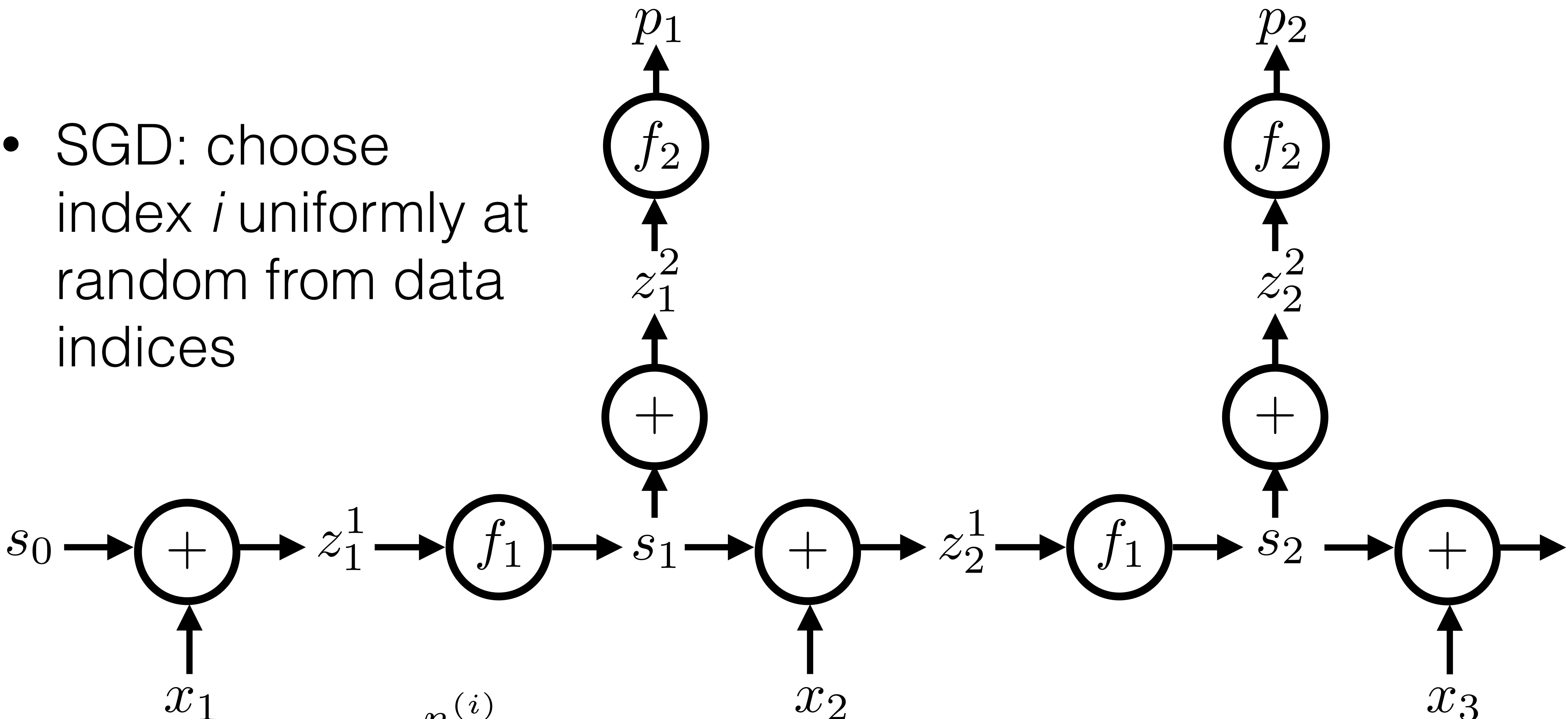
$$L_{\text{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$

$$L_t := L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$

$$\frac{dL_{\text{seq}}(p^{(i)}, y^{(i)})}{dW^{sx}} = \sum_{t=1}^{n^{(i)}} \frac{dL_{\text{elt}}(p_t^{(i)}, y_t^{(i)})}{dW^{sx}}$$

RNNs: a taste of backpropagation

- SGD: choose index *i* uniformly at random from data indices



$$L_{\text{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$

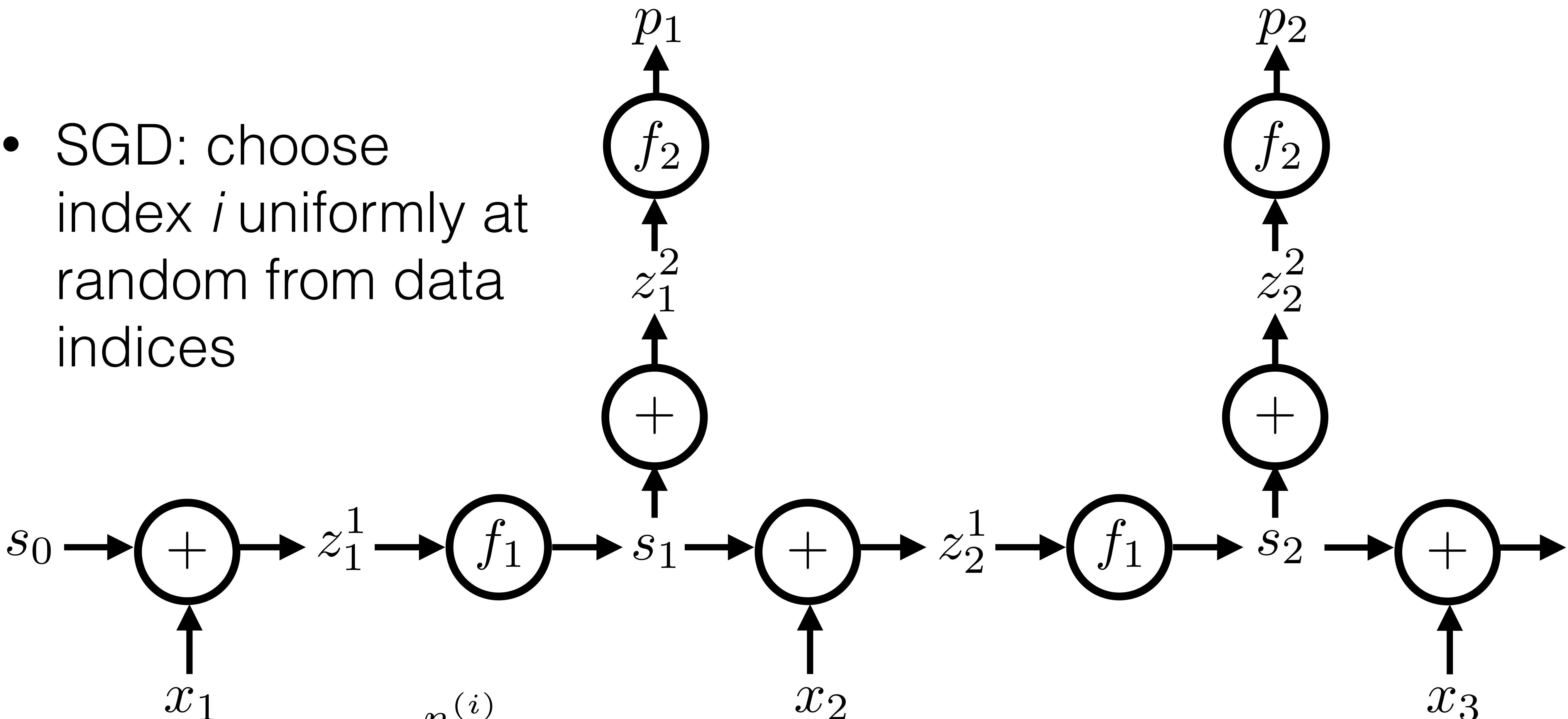
$$L_t := L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$

$$\frac{dL_{\text{seq}}(p^{(i)}, y^{(i)})}{dW^{sx}} = \sum_{t=1}^{n^{(i)}} \frac{dL_{\text{elt}}(p_t^{(i)}, y_t^{(i)})}{dW^{sx}}$$

- Need: $\dfrac{dL_t}{dW^{sx}}$

RNNs: a taste of backpropagation

- SGD: choose index *i* uniformly at random from data indices



$$L_{\text{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$

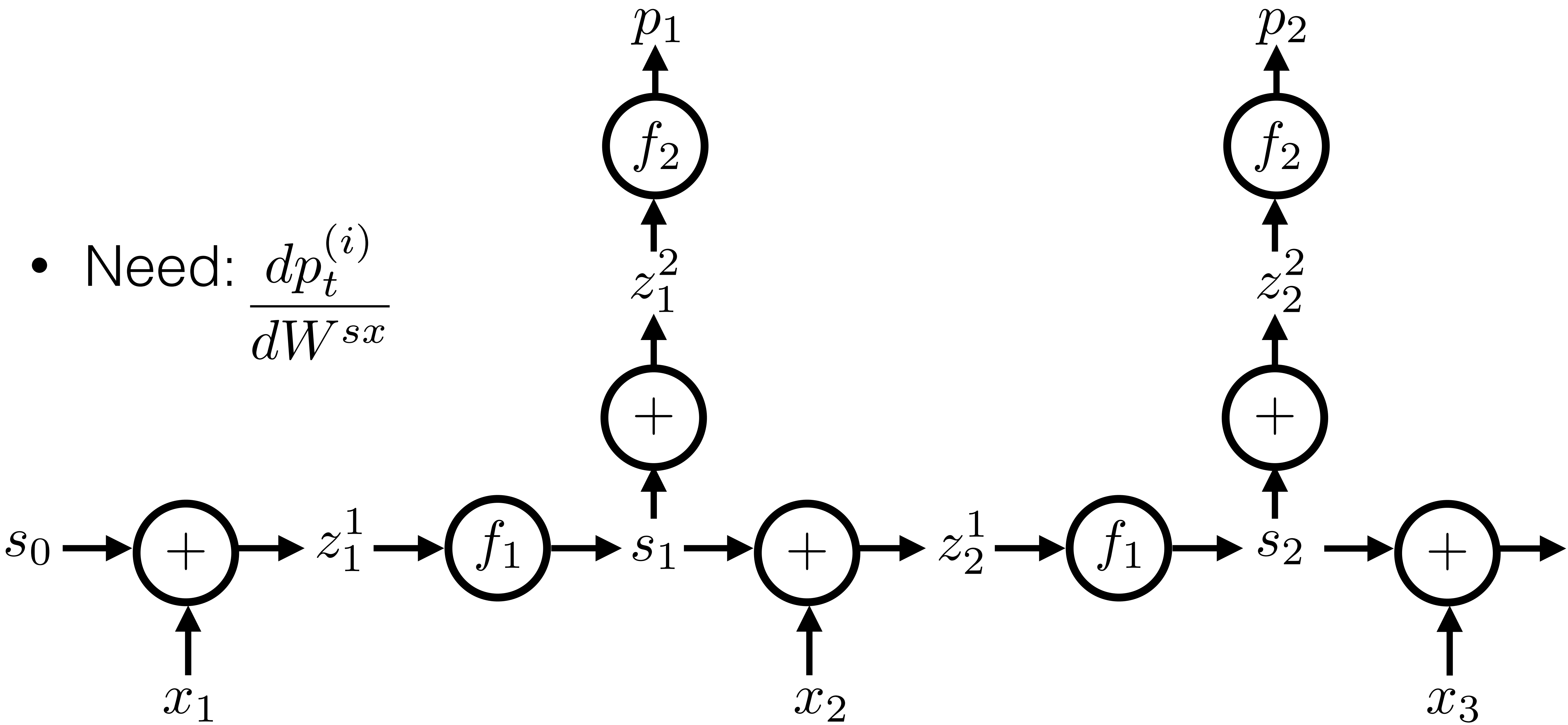
$$L_t := L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$

$$\frac{dL_{\text{seq}}(p^{(i)}, y^{(i)})}{dW^{sx}} = \sum_{t=1}^{n^{(i)}} \frac{dL_{\text{elt}}(p_t^{(i)}, y_t^{(i)})}{dW^{sx}}$$

- Need: $\dfrac{dp_t^{(i)}}{dW^{sx}}$

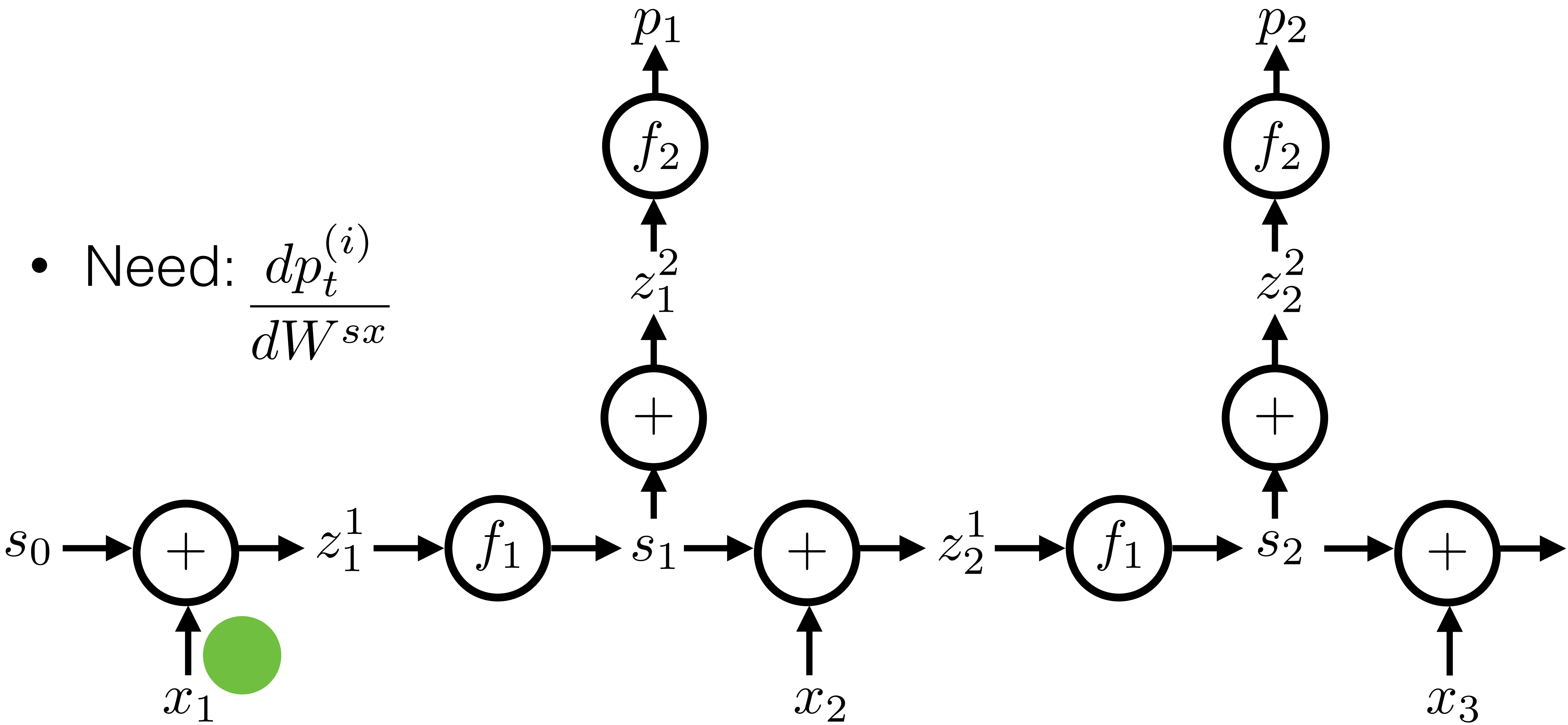
RNNs: a taste of backpropagation

- Need: $\dfrac{dp_t^{(i)}}{dW^{sx}}$



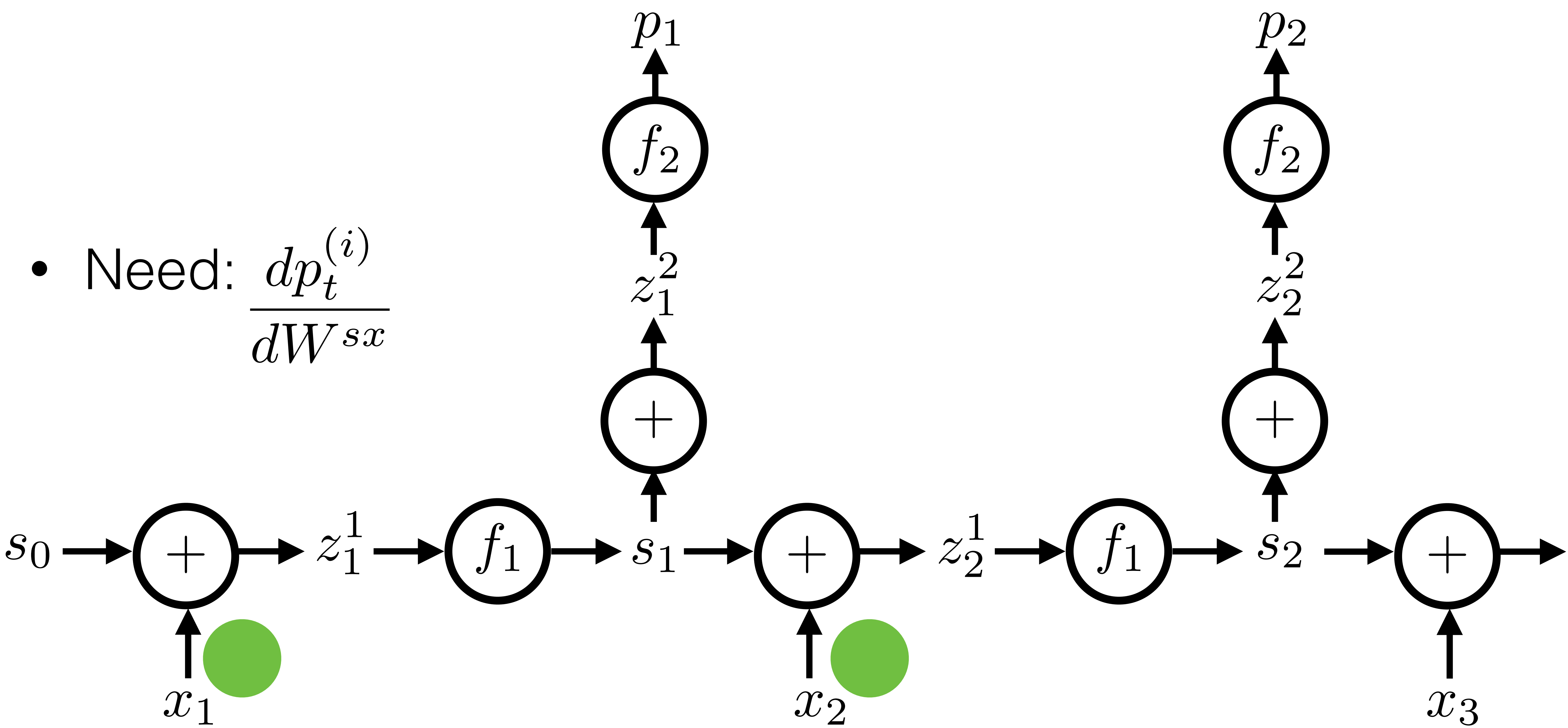
RNNs: a taste of backpropagation

- Need: $\frac{dp_t^{(i)}}{dW^{sx}}$



RNNs: a taste of backpropagation

- Need: $\dfrac{dp_t^{(i)}}{dW^{sx}}$



RNNs: a taste of backpropagation

- Need: $\dfrac{dp_t^{(i)}}{dW^{sx}}$