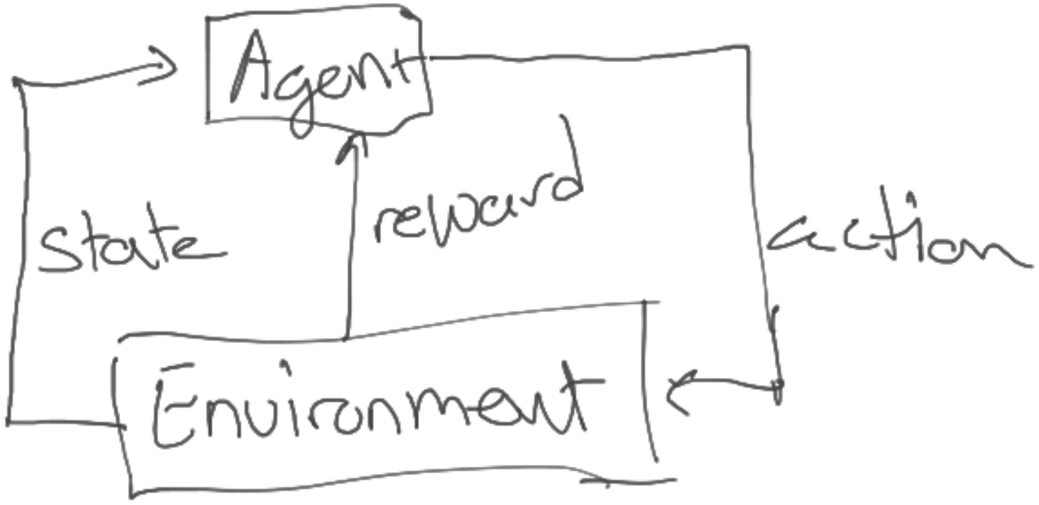
What is reinforcement learning (RL)?

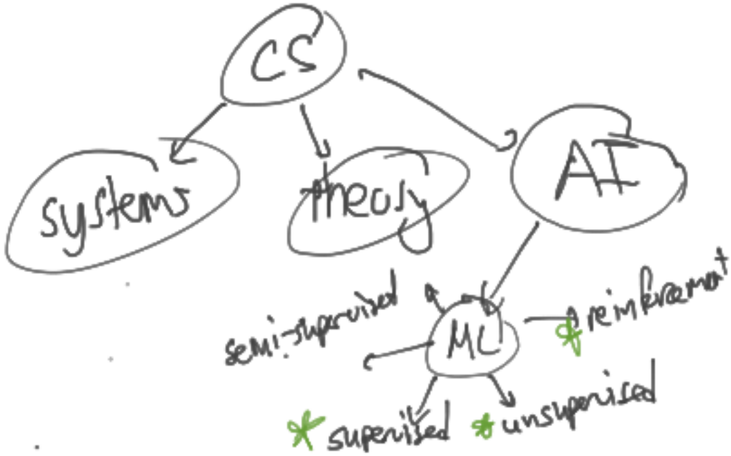
"Reinforcement learning is an area of machine learning, inspired by behaviorist psychology, concerned with how an agent can learn from interactions with an environment."

– Wikipedia / Sutton & Barto / Phil

1998

Agent - Environment Diagram





Neuroscience: How do animals learn?
A specific agent or set of agents.

- The study of some examples of learning and intelligence.

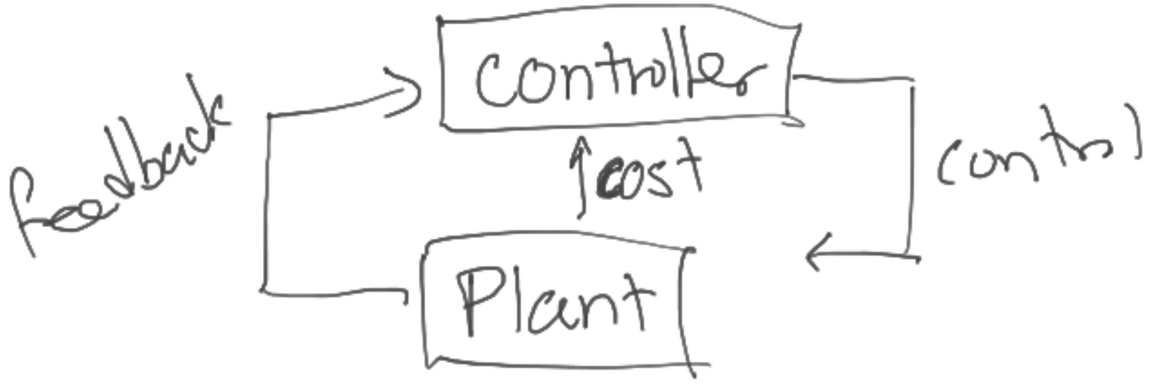
(ML)
RL : How can we make an agent that learns?

- The study of learning & intelligence. (in general)
(animal or not),

Dopamine ≈ Temporal difference error.

Two most related fields

- Operations research
- Control (adaptive / classical)

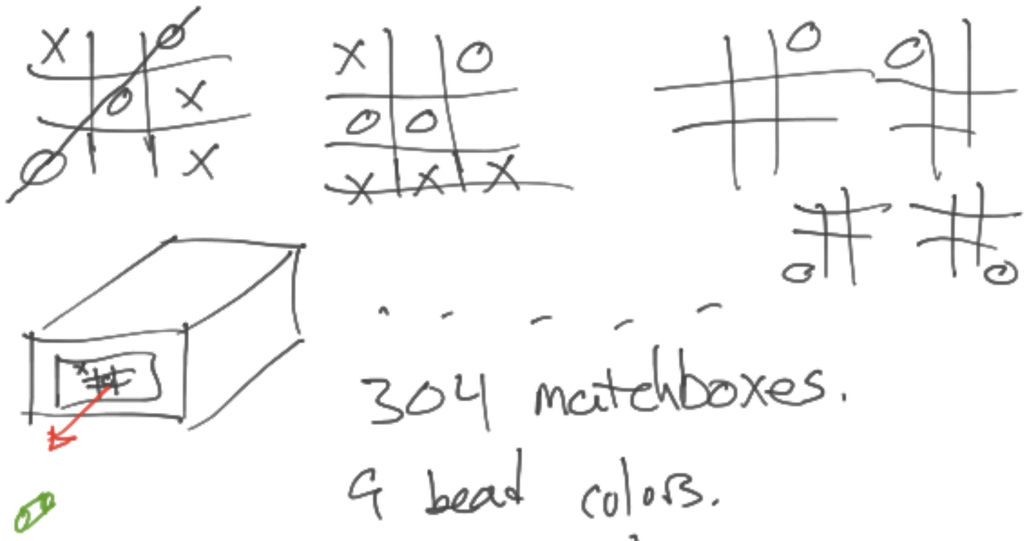


- Main difference is that these fields typically assume the environment (Plant) can and should be directly approximated.

Key properties:

1) Evaluative feedback (not instructional)
   ↳ "This is how good the outcome was," not "this is what you should have done."
2) Sequential.
   ↳ * no data set! *

MENACE (Donald Mitchie, 1961)



304 matchboxes.

4 bead colors.

6 possible moves.

Win: Put back bead, +3 more of same color.

Lose: Remove beads.

Tie: (Draw) Put back bead, +1 more of same color.

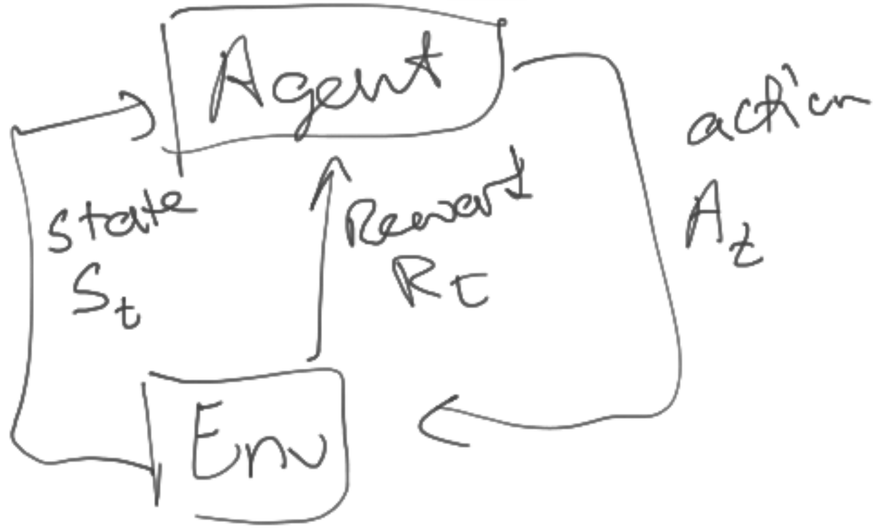
Idea:

- Act at least slightly random.
- Good outcome
  - to take chosen actions more often
- Bad outcome
  - to take chosen actions less often.

Act randomly ???

- "Necessary" for learning.
  - really, exploration is necessary, and randomness is one way to explore
- "exploration" vs "exploitation".
  - exploration: taking actions that think are suboptimal
  - exploitation: choosing actions think are best.

CMPSCI 687



t: time step.
$t \in \{0, 1, 2, \ldots\}$

S_t: State at time t.
(Agent's observation of state)

A_t: Action at time t.

R_t: Reward at time t.
$R_t \in \mathbb{R}$

A policy π is a way of selecting actions.

$\pi(s,a) = Pr(A_t = a \mid S_t = s)$

~~$A_t = \pi(S_t)$~~

- many policies - some good some bad.

Agent's goal: find a policy that maximizes the expected amount of reward the agent receives.

↳ not deterministic!

Objective function: $J(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t ; \pi\right]$

↳ Means "assuming R_t generated using π"

- Could be infinite! → $J(\pi)$
- Doesn't discount based on time

$\gamma \in [0, 1]$ ← YAHP ↳ yet another hyperparameter.

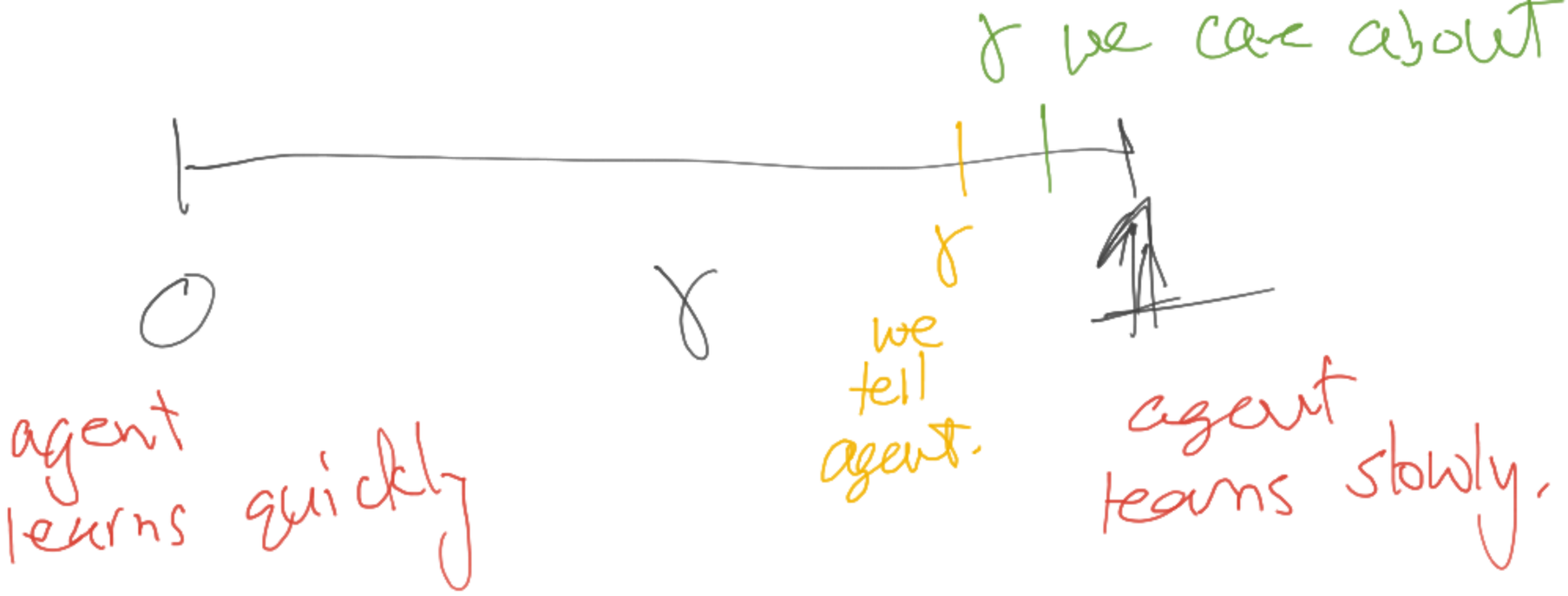
γ → care about short term

γ → care about long term.

Typically $\gamma \approx .9 \leftrightarrow 1$

$\gamma^0 R_0 + \gamma^1 R_1 + \gamma^2 R_t \ldots \ldots$

$1 R_0 + .5 R_1 + .25 R_t + \ldots \ldots$



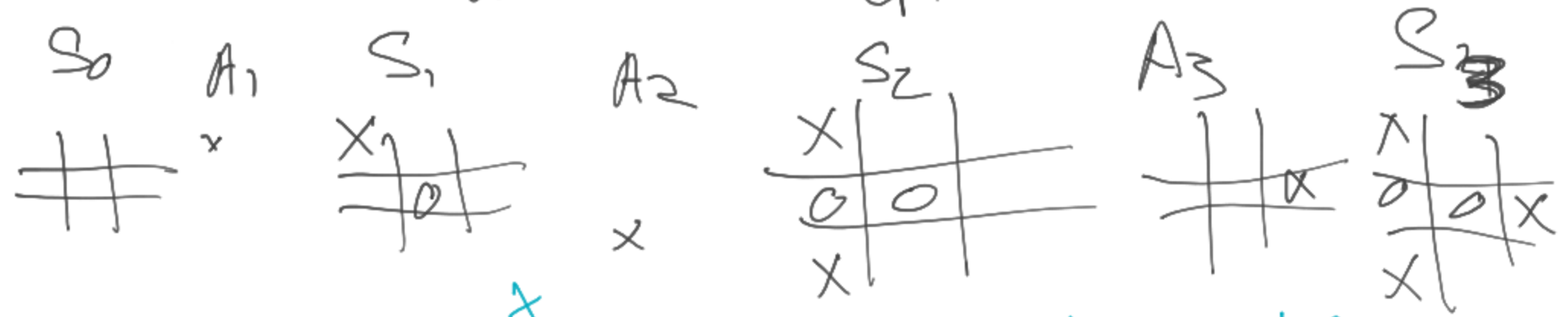
Agent's goal is to find or approximate

$$\pi^* \in \arg\max_{\pi} J(\pi)$$

π^* is called an optimal policy.

"Independent" trials: each starting from $t=0$, are called "episodes."

one episode: $S_0, A_1, S_1, A_2, S_2, A_3, S_3$



next episode.

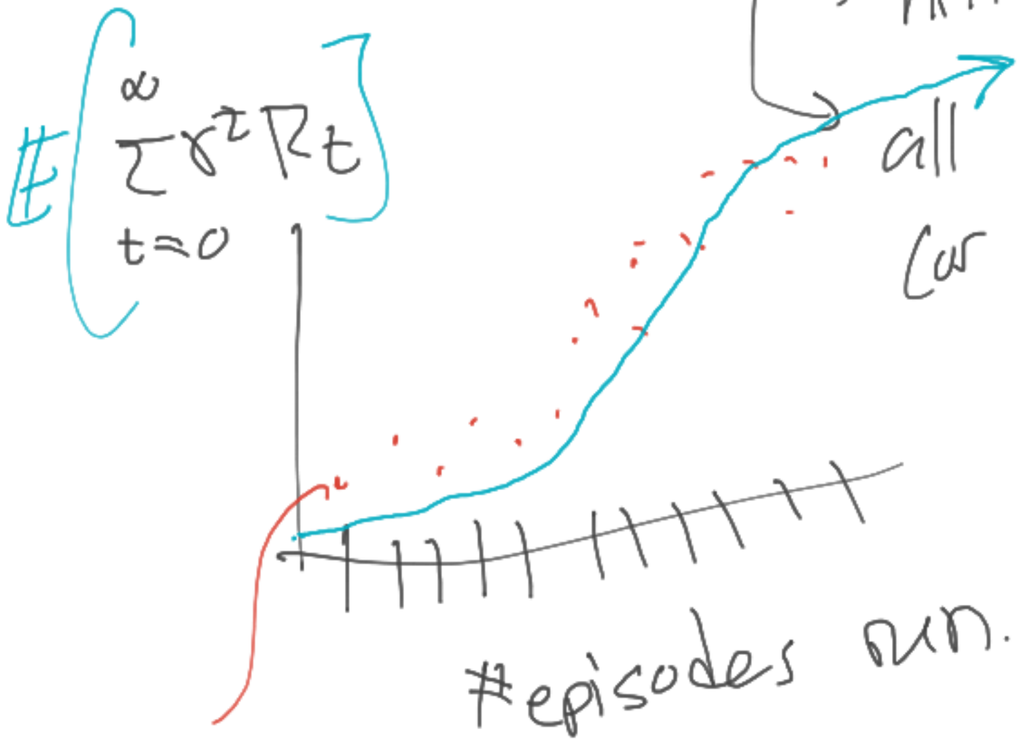
The dynamics of env do not change across trials, e.g. actions in one episode do not impact environment in later episodes.

- Way agent acts (agent's policy) can change across episodes.

$(S_t, A_t, R_t)_{t=0}^{\infty}$ independent rand variables across episodes.

Episodes end when the environment reaches a state called a "terminal state"

(↳ hill top in Mountain Car, all cells filled Car someone won) in Tic-Tac-Toe



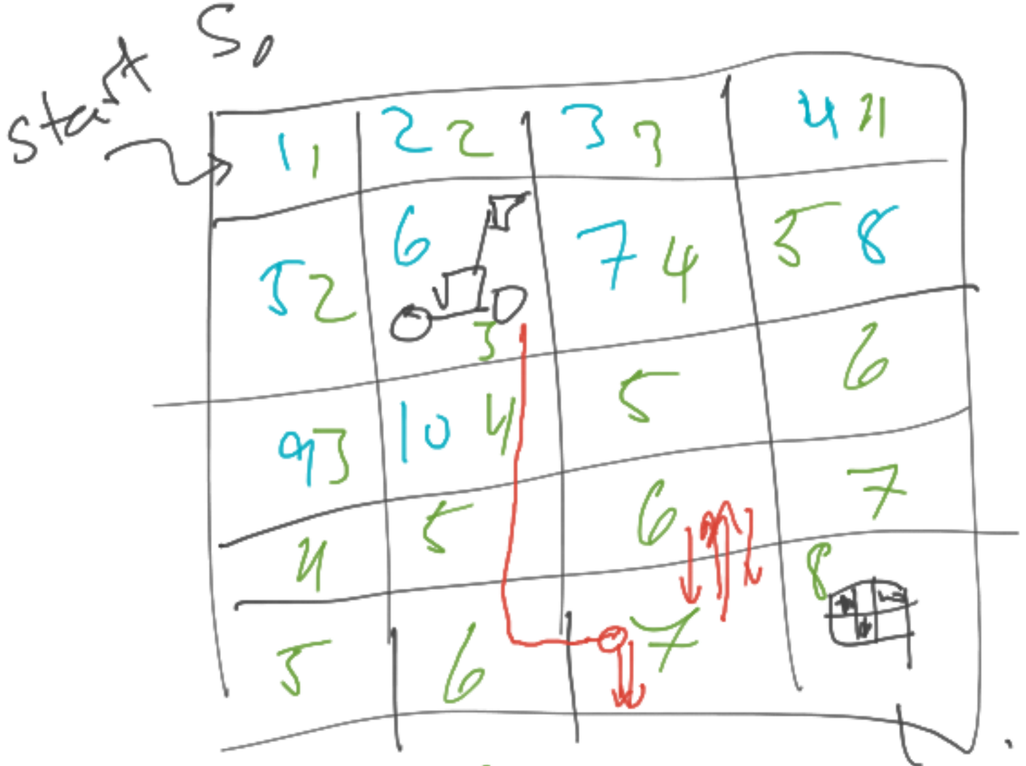
Flow:

$(S_0, A_0, R_0, S_1, A_1, R_1, \ldots)$

→ For all times after a terminal state, $R_t = 0$.

Gridworld:

$A_t \in \{\text{up}, \text{down}, \text{left}, \text{right}\}$



$S_t = (2,4)$

$S_t = 6$

$A_2 = \text{right}$

$R_t =$

$S_{t+1} = (3,4)$

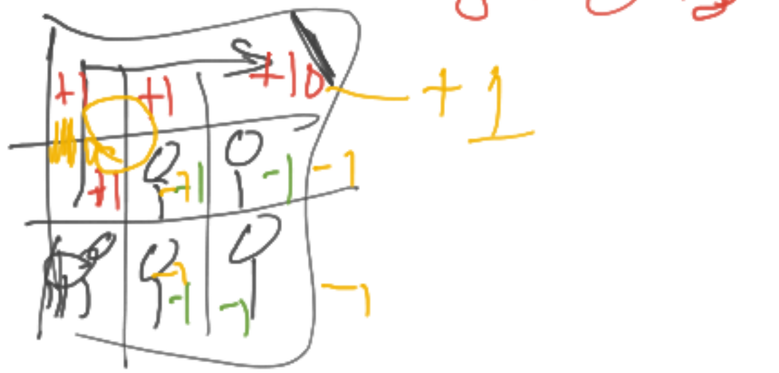
$R_t = 4$

$t = 10$
$t = 20$

R_t design flower bed

How to define R_t is yet another hyperparameter.

$\gamma = 0.5$

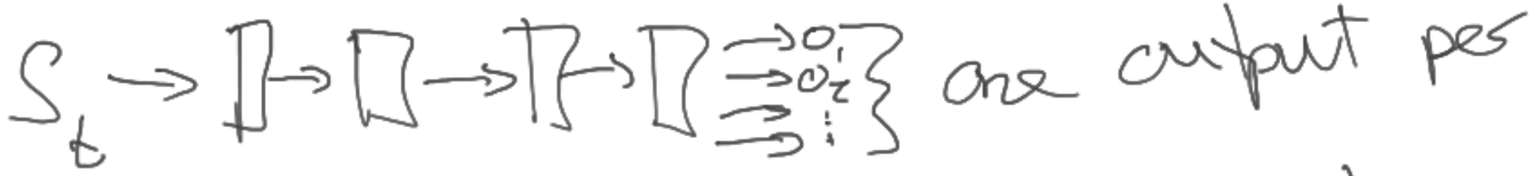


→ Give rewards for the outcomes you want, not for how you think the agent should achieve the outcome.

"reward shaping" Andrew Ng.

How represent π?

- Simplifying assumption: A_t has a small finite number of possible values.



one output per possible action.

Softmax to select the action:

$$\pi(s,a) = Pr(A_t = a \mid S_t = s) = \frac{e^{o_a}}{\sum_i e^{o_i}}$$

Supervised learning

RL

model → policy

w → θ

"model parameters" → "policy parameters"

f_w → π_θ

↳ stochastic, like classification (uses softmax)

↓

deterministic for regression.

ℓ → $-J$

$$\ell(w_k) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_{w_k}(x_i))^3$$

$$\frac{\partial \ell(w_k)}{\partial f_{w_k}(x_i)} = \frac{1}{n}\sum_{i=1}^{n} 3(y_i - f_{w_k}(x_i))^2 \frac{\partial}{\partial f_{w_k}(x_i)}(y_i - f_{w_k}(x_i))$$

$$\frac{\partial}{\partial f_{w_k}(x_i)}(y_i - f_{w_k}(x_i)) = -1$$

$$\frac{\partial \ell(w_k)}{\partial w_{kj}} = \frac{\partial \ell(w_k)}{\partial f_{w_k}(x_i)} \frac{\partial f_{w_k}(x_i)}{\partial w_{kj}}$$

$$= -\frac{3}{n}\sum_{i=1}^{n}(y_i - f_{w_k}(x_i))^2 x_{ij}$$

$$f_{w_k}(x_i) = \sum_{\beta} w_{k\beta} x_{i\beta}$$

$$\frac{\partial f_{w_k}(x_i)}{\partial w_{kj}} = \frac{\partial}{\partial w_{kj}} w_{kj} x_{ij} = x_{ij}$$

1c

$$f_{w_k}(x_i) = \sum_\beta w_{k\beta}\, x_{i\beta}$$

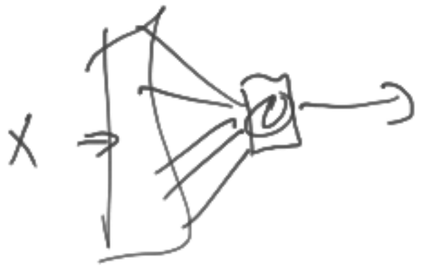
$$\frac{\partial f_{w_k}(x_i)}{\partial w_{kj}} = \frac{\partial}{\partial w_{kj}} \sum_\beta w_{k\beta}\, x_{i\beta}$$

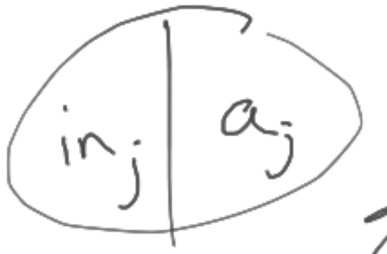
$$= \sum_\beta \frac{\partial}{\partial w_{kj}} w_{k\beta}\, x_{i\beta} \qquad j=2 \quad \beta=1$$

$$= \frac{\partial}{\partial w_{kj}} w_{k1}\, x_{i1} + \frac{\partial}{\partial w_{kj}} w_{k2}\, x_{i2} + \frac{\partial}{\partial w_{kj}} w_{k3}\, x_{i3} \cdots$$

$j=2$ $\qquad$ $=0$ $\qquad$ $=x_{i2}$ $\qquad$ $=0$

$$= x_{i,2}$$





linear model
weight, divided by 7.4.

$$in_j = \sum_i w_{ij} x_i$$

$$a_j = \sigma(in_j)$$

$$a_j = 7.4\, in_j$$

$$\frac{\partial\, \sigma(in_j)}{\partial in_j} = \sigma(in_j)(1-\sigma(in_j))$$

≤ 0.5

$$\frac{\partial\, \sigma(\sigma(in_j))}{\partial in_j} = \sigma(\sigma(in_j))(1-\sigma(\sigma(in_j)))\,\sigma(in_j)(1-\sigma(in_j))$$

≤ 0.5

$in_j = a x_{i1} + b x_{i2}$

$a_j = in_j = a x_{i1} + b x_{i2}$

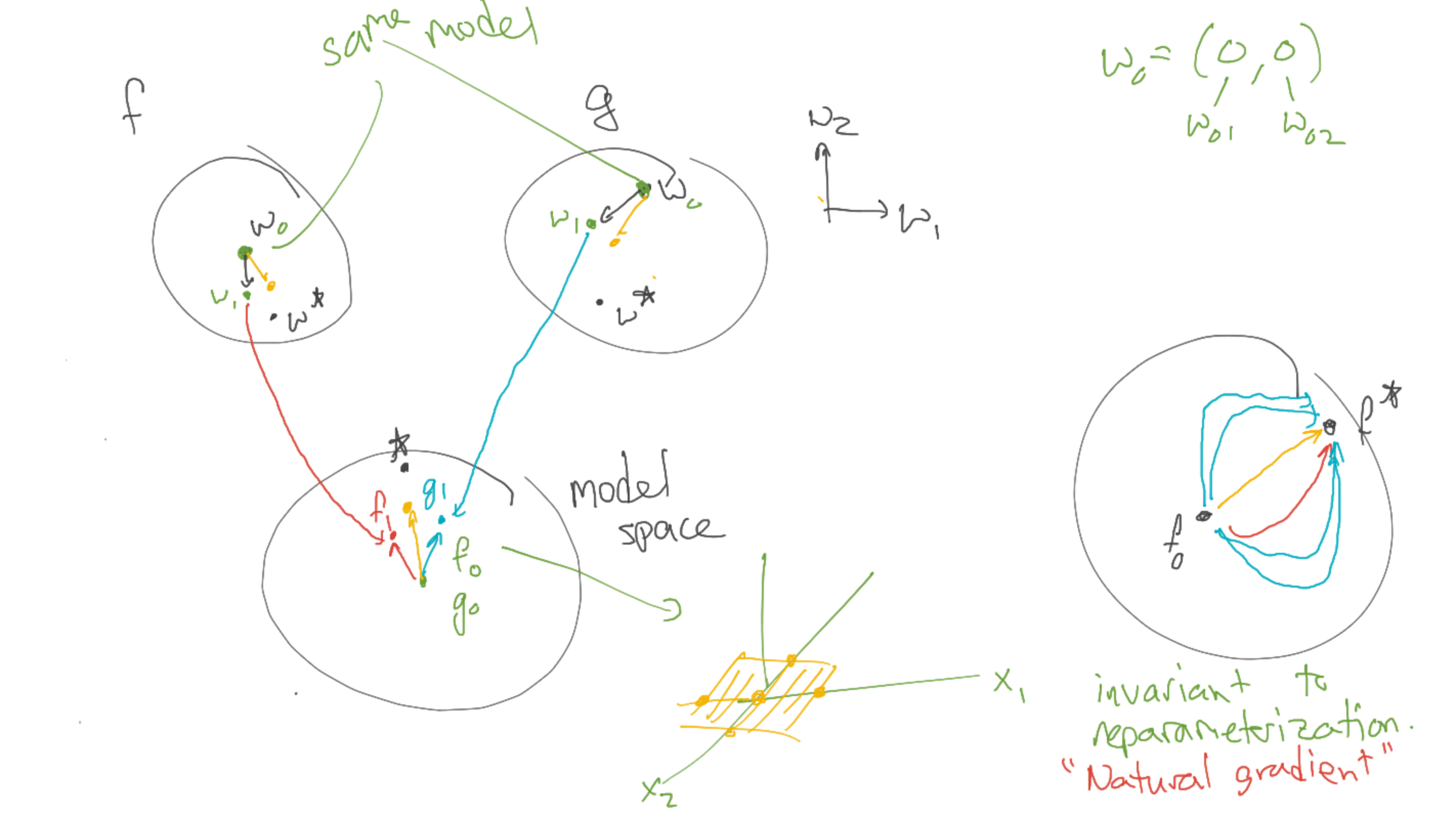
$= ea x_{i1} + eb x_{i2} + fc x_{i1} + fd x_{i2}$

$= x_{i1}(ea + fc) + x_{i2}(eb + fd)$

$f_{w_k}(x_i) = w_{k1} x_{i1} + w_{k2} x_{i2}$

w_{k1} → $ea + fc$

w_{k2} → $eb + fd$


same model
f
g
w_0
w_1
w^*
w_2
w_1
model space
f_1
g_1
f_0
g_0
x_1
x_2
$w_0 = (0, 0)$
w_{01} w_{02}
f^*
f_0
invariant to reparametrization.
"Natural gradient"

$x_{11} = 0$

$x_{12} = 1$

$y_1 = 1$

→ one step of gradient descent on loss.

$x_{21} = 0$

$x_{22} = 0$

$y_2 = 0$