QFL

Quioto Forest Legends

A B C

0 1 2 3

door opens → one combatant per side enters → fight over gems → bell rings 1-4 times → awaken Leafstache? 24 rings → feed Bonsly? at least 10 at least every 4 times in hut → door opens

gems	rings
20	
20	1
20	2
20+3	5
45 ~~20+25~~	9
45+5	11
91 ~~45+46~~	15
91+5	17
91+2	20
91+2	21
115 ~~91+24~~	25

Card images by eBay user furret_inc https://www.ebay.com/itm/OLD-Original-Vintage-Pokemon-10-Card-LOT-1st-Edition-Holo-Rare-Played/233176095585?hash=item364a60db61:g:3OQAAOSwG7NclWhm
Except Bonsly original art by Ken Sugumori for Pokemon Diamond and Pearl downloaded from Bulbapedia https://bulbapedia.bulbagarden.net/wiki/File:438Bonsly.png
Treebeard from The Lord of the Rings (1978) via Wikimedia https://en.wikipedia.org/wiki/File:BakshiTreebeard.JPG
Used under Fair Use Doctrine
Kobe Bell by Joe Mabel, CC BY-SA 3.0 <http://creativecommons.org/licenses/by-sa/3.0/>, via Wikimedia Commons https://upload.wikimedia.org/wikipedia/commons/8/89/Seattle_Center_-_Kobe_Bell_02A.jpg

OFFENSE

		Hitmonchan (3 rings except as noted)	Machamp	Magneton	Mewtwo
Chansey	1-4	2	5	19	0
	5	8	12	-100	4
	6-10	5	5	22	-100
	11-12	-1	-100	47	47
	13-20	3	0	0	0

4 rings
1 ring
otherwise 2 rings

DEFENSE

Charizard	1-4	1	4	11	48
	5	14	32	-100	-9
	6-10	2	5	14	41
	11-12	-4	0	-9	-100
	13-20	2	0	0	0
Clefairy	1-4	1	21	-7	53
	5	4	-4	8	-100
	6-10	2	0	15	-11
	11-12	1	46	-8	-100
	13-20	-3	5	18	0

states encapsulate (yards to go , downs left , yards to go to 1st down , ticks left)
gems to collect — or min until Bonsly — choc covered gems needed — or bell rings left

wakes

possible values 101 , 5 , 99 , 25

total # of states ~1.25 million

$V(s)$ = P(offense wins in state s)

defense

offense	A	B	C
0	–	–	–
1	–	–	–
2	–	–	–
3	–	–	–

matrix for each state

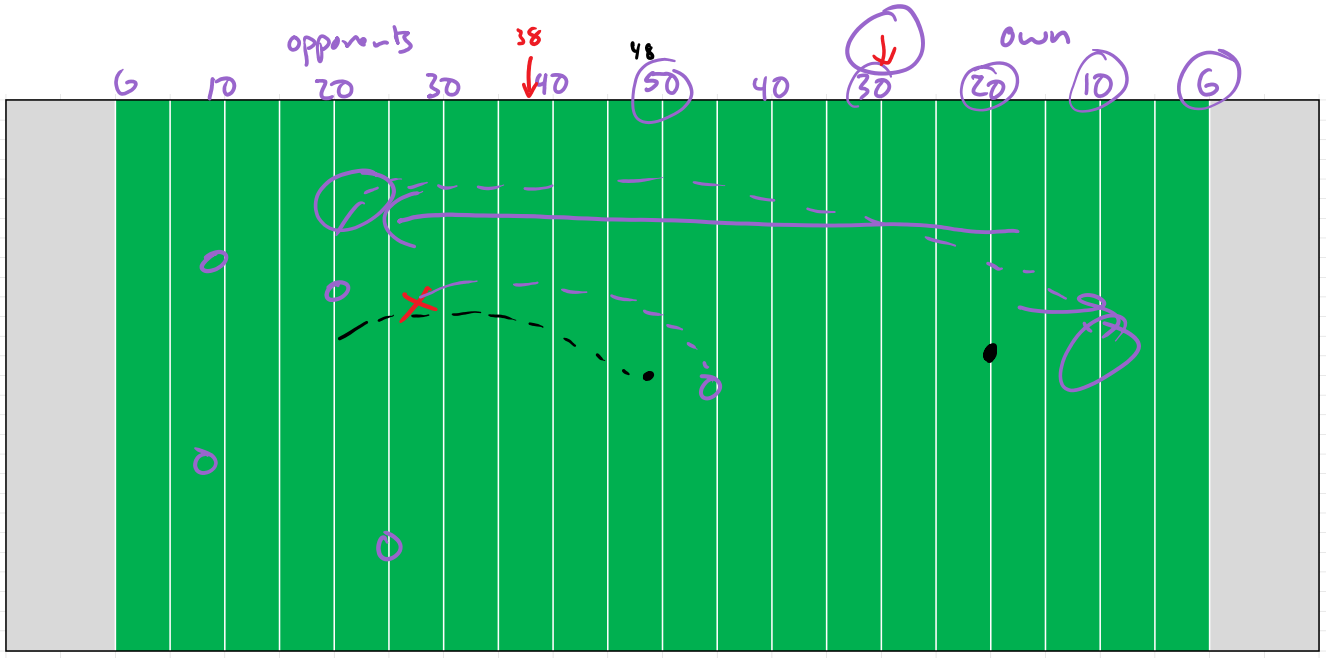
fill matrix

$\sum_{s \to s'} p(s \to s') \cdot (r(s \to s') + v(s'))$

use l.p. to get $v(s)$

$r(s \to s')$: -1 if s' losing, +1 if s' winning, 0 otherwise

compute from terminal → initial



Home	quarter	time	down	distance	Away
13	4	2:00	1	10	17

Q Learning

$q(s,a)$ = value of taking action a in state s

$V(s) = \max_a q(s,a)$

initialize $q(s,a) = 0$ for all s,a

while not done

$s \leftarrow s_0$ initial state

while s not terminal

choose action a — ε-greedy: prob ε: pick random action; $1-\varepsilon$: pick a to max $q(s,a)$

observe transition (s, a, r, s')

update $q(s,a) \leftarrow q(s,a) + \alpha\left(r + \gamma \cdot \max_{a'} q(s',a) - q(s,a)\right)$

$s \leftarrow s'$

α: learning rate ↓ as you update each $q(s,a)$

Function Approximators

Linear Approximator → learn an approximation of $q(s,a)$

Define features of state/action pairs

$f_1(s,a) = \begin{cases} 1 & \text{if short yardage} \\ 0 & \text{otherwise} \end{cases}$

$f_2(s,a) = \begin{cases} 1 & \text{if desperate} \\ 0 & \text{otherwise} \end{cases}$

$f_3(s,a)$

$\vdots$

$f_4(s,a)$

fxns of just state

$f_{1a}(s,a') = \begin{cases} f_1(s,a) & \text{if } a = a' \\ 0 & \text{otherwise} \end{cases}$

$\hat{Q}(s,a) = w_1 * f_1(s,a) + w_2 * f_2(s,a) + \cdots + w_n * f_n(s,a)$

for nonterminal s

$q(s,a) = 0$ if s is terminal

need to learn weights

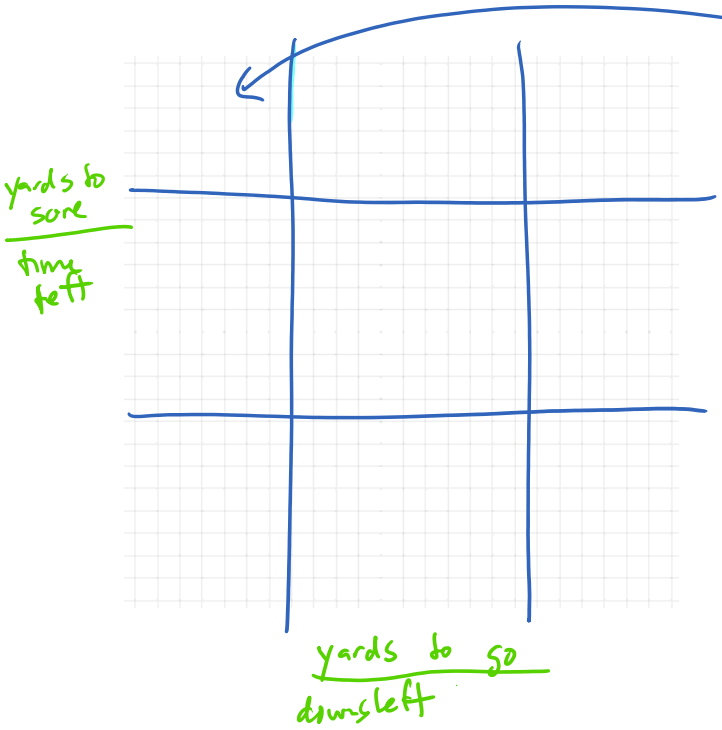
In state s

Choose action a

Observe transition (s, a, r, s')

Update for each feature $w_i \leftarrow w_i + \alpha \left(r + \gamma \max_{a'} \hat{q}(s',a') - \hat{q}(s,a) \right) \cdot f_i(s,a)$

State Aggregation (Coarse Coding/Buckets/Grid)



$$f_{1,1}(s,a) = \begin{cases} 1, & \text{if } s \text{ in grid } 1,1 \\ 0 & \text{otherwise} \end{cases}$$

and copy for each action

$$f_{r,c,a}(s,a') = \begin{cases} 1 & \text{if } s \text{ in cell } r,c \text{ and } a=a' \\ 0 & \text{otherwise} \end{cases}$$

one weight update per step
(for weight $w_{r,c,a}$ on feature $f_{r,c,a}$)

while not done
  $s \leftarrow s_0$
  while s not terminal
    choose action a
    observe transition (s, a, r, s')
    determine grid cells (bucket) $\hat{s}$ and $\hat{s}'$ that s and s' belong to
    update $\hat{q}(\hat{s},a) \leftarrow \hat{q}(\hat{s},a) + \alpha\left(r + \gamma \cdot \max_{a'} \hat{q}(\hat{s}',a') - \hat{q}(\hat{s},a)\right)$
    $s \leftarrow s'$

these are the $w_{r,c,a}$, where $\hat{s}$ = the r,c that s is in

weights on the features $\equiv \hat{q}(\hat{s},a)$ for the corresponding superstates (buckets) and actions