Function Approximators

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

Define features of state/action pairs

$f_1(s,a)$ = { 1 if short yardage, 0 otherwise
$f_2(s,a)$ = { 1 if desperate, 0 otherwise
$f_3(s,a)$
⋮
$f_4(s,a)$

fxns of just state

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

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

for nonterminal

need to learn weights

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

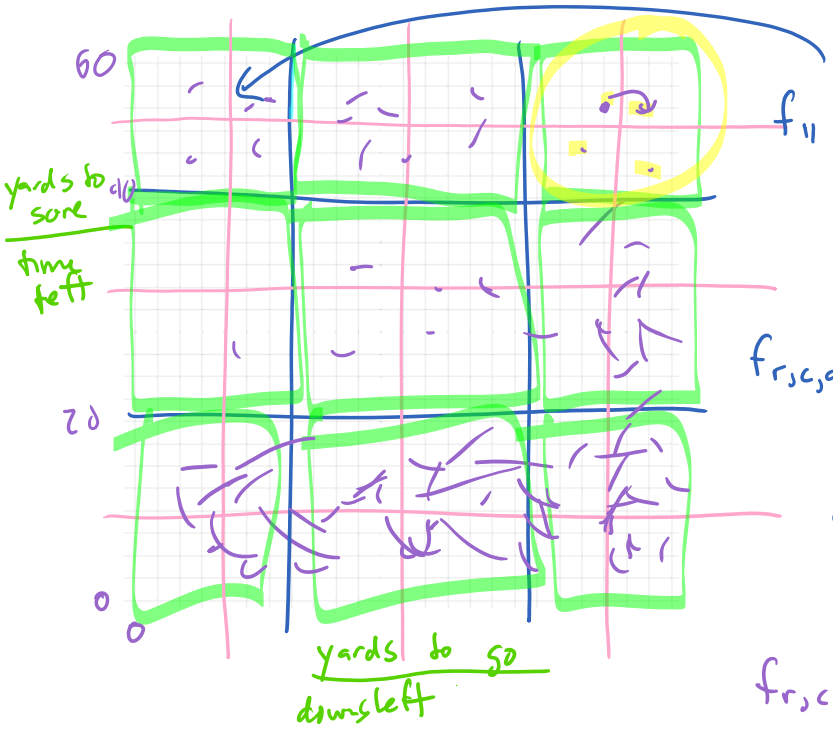
In state s

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)



$$f_{11}(s,a) = \begin{cases} 1 & \text{if } s \text{ in grid } 1,1 \\ 0 & \text{otherwise} \end{cases}$$

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 $f_{r,c,a}$)

$f_{r,c,a}(s,a') = 1$ for the r,c that s is in and $a=a'$
0 for all other r,c,a

$\hat{q}(s,a) = w_{r,c,a}$

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

$\hat{q}(s,a)$

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

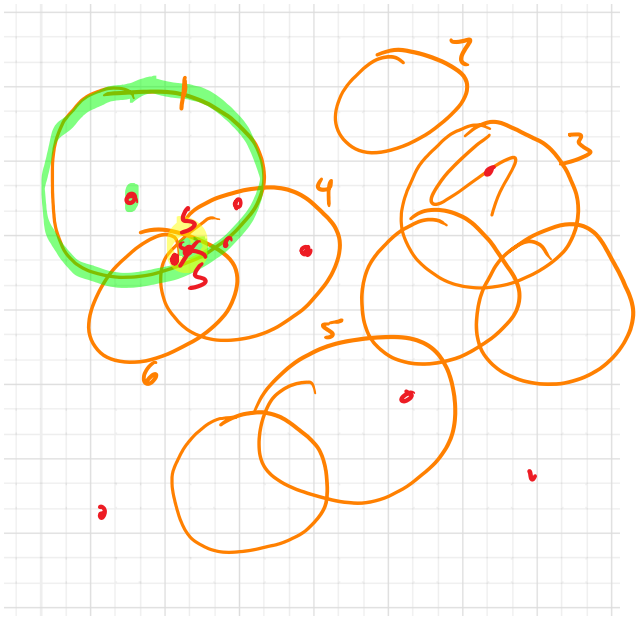
weights on the features $\equiv \hat{q}(\hat{s},a)$ for the corresponding superstates (buckets) and actions

sparse rewards – rewards on very few steps

QFL – rewards only on steps into terminal states

reward shaping – introduce artificial rewards for progress towards real rewards

State Aggregation (Coarse Coding/Buckets/Grid)



Coarse Coding

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

(make copy for each action)

$f_1(s,a) = 1$
$f_2(s,a) = 0$
$f_3(s,a) = 0$
$f_4(s,a) = 1$
$f_5(s,a) = 0$
$f_6(s,a) = 1$

sparse rewards – rewards on very few steps

QFL – rewards only on steps into terminal states

reward shaping - introduce artificial rewards for progress towards real rewards

Classifiers

Regression: function — attributes → value — $g(s,a)$

GFL: 4 components of state action

Classifier: function — attributes → class → species of iris / digit (0, ..., 9)

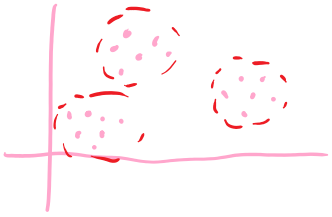
state — action

Iris flower data set

MNIST database http://yann.lecun.com/exdb/mnist/

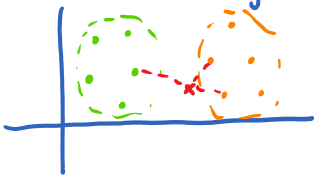
Learning: supervised – have examples, generalize from those
↳ grandmaster database

reinforcement – observe transitions, earned rewards



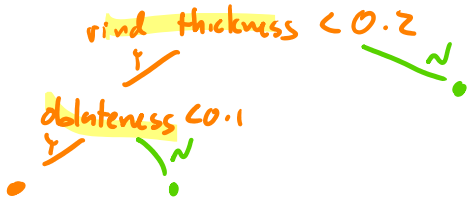
unsupervised – discover unknown relationships

Methods: k-nearest neighbors

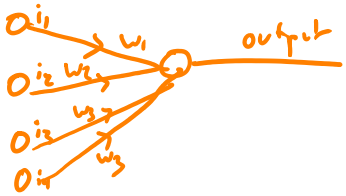


2 of 3 nearest are • so predict •

decision trees
nested ifs



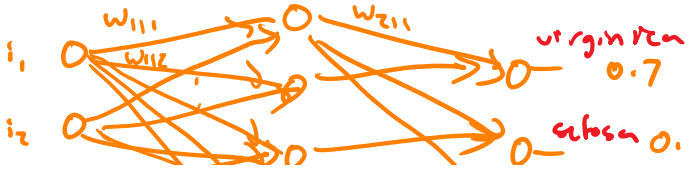
perceptron

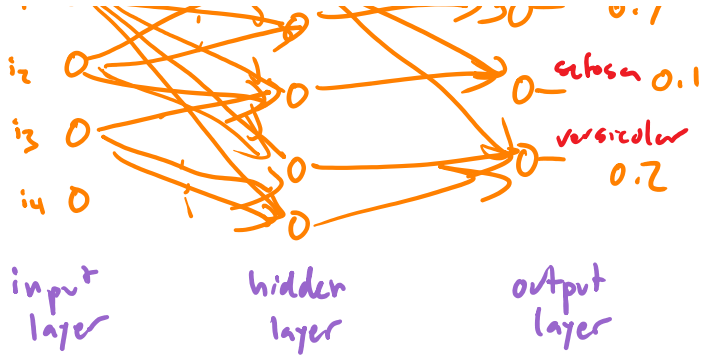


$f\left(\sum_{j=1}^{k} i_j \cdot w_j\right)$

↳ $f(x) = \begin{cases} 1 & \text{if } x \geq b \\ 0 & \text{otherwise} \end{cases}$

multi-layer perceptron (artificial feed-forward neural network)




i2
i3
i4
setosa 0.1
versicolor
0.2
input layer
hidden layer
output layer

Input Representation

Input / Output

categorical

versicolor = 0
virginica = 1
setosa = 2
yalica = 3

versicolor: 0

50/50 versicolor/virginica	75/25 versicolor/virginica	50/50 virginica/setosa	50/50 versicolor/setosa	~~33/33/33~~
0.5	0.25	1.5	2.5	

	versicolor	virginica	setosa
one-hot	1	0	0
	0	1	0
	0	0	1

date

seconds since epoch (normalized)

Jan 1 1900 =
Jan 1 1999 =

month / day / year

Jan 1 1900 =	1	1	1900
Jan 1 1999 =	1	1	1999
Jan 31 2020	1	31	2020
Feb 1 2020	2	1	2020
Dec 31 1998	12	31	1998

fuzzy

Jan	Feb	Mar	...								Dec
0	0	0	0	0	0	0	0	0	0	0	0

Jan 1 .48
Jan 10 .3
Mar 15
Nov 16
Dec 27

rolls

1	1	3	4	6
1	1	1	2	5