Roadmap

Understanding Alpha Go / Alpha Go Zero

- ✓ MCTS
- ✓ reinforcement learning
- supervised learning
- ✓ neural networks
- convolutional neural networks

Classifiers

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

Classifier: function — attributes (position) → class (what move to make)

Iris flower data set

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

Learning:

- reinforcement – observe state transitions, rewards
- supervised – examples; generalize from those examples (games – examples of expert play)
- unsupervised – discover unknown relationships / groupings

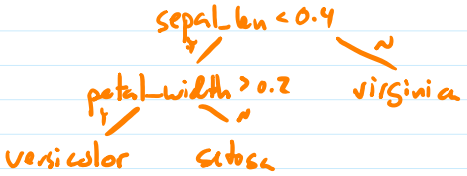
Methods:

k-nearest neighbors

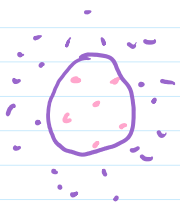
2 of 3 nearest are • so classify as •

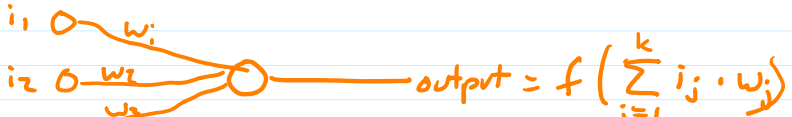
decision trees

nested ifs



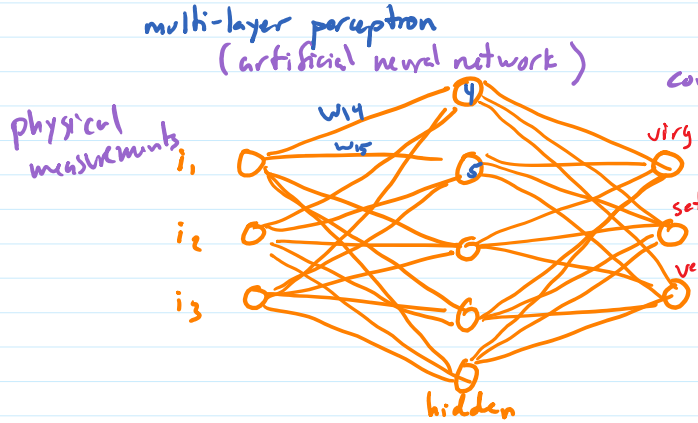
perceptron





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

multi-layer perceptron (artificial neural network)



confidence on each class

virg	.9	.6
set	0.04	or .4
ver	0.07	.0

most likely virginica

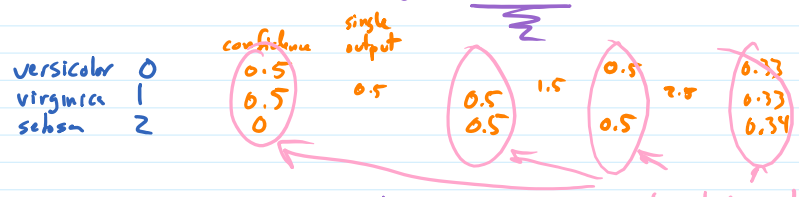
probably virginica but can't rule out setosa

Input Representation

Input / Output – numeric!

categorical

if inputs are numeric, use as is (but normalize)



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

single output can't capture confidence in each class, so one output for each class giving confidence in that class

Jan 1 1970

date — seconds since epoch (normalized)

Jan 1 1900 = −1000000
Jan 1 1999 = 400000

these need to be close in order to generalize from Jan 1 1900 to Jan 1 1999

3 inputs

month/day year

Jan 1 1900 = 0/12 0/31 (1900−1900)/200
Jan 1 1999 = 0/12 0/31 (1999−1900)/200
Dec 31 1950 = 11/12 30/31 (1950−1900)/200

→ to generalize from Jan 1 to Dec 31, need their representations to be close

fuzzy

	Jan						...	Dec
	0	0	0	0	0	0	...	0
Jan 1	.51							.49
Mar 15			1.0					
Dec 31	.49							.51
Feb 7	.3	.7						
July 15					1.0			

kinda close

close together!

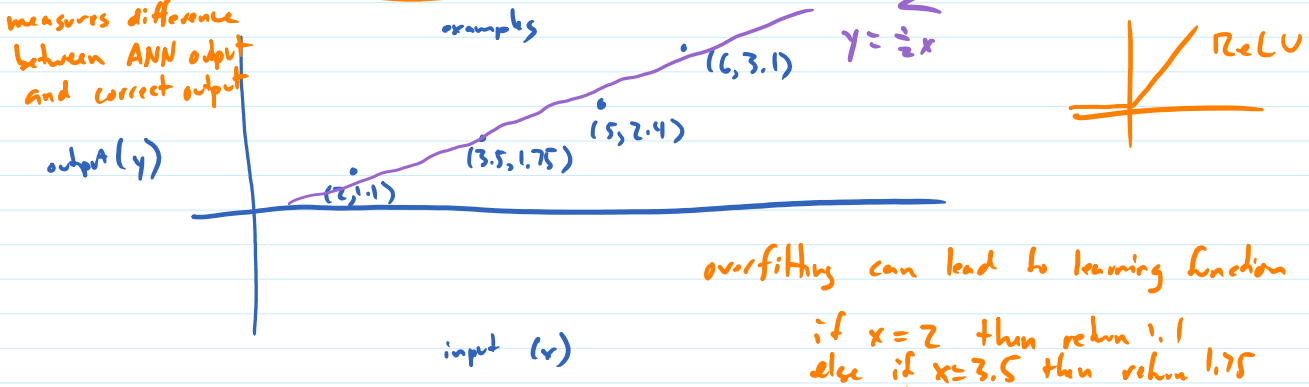
ANN Supervised Learning

initialize weights randomly

split dataset into
- training data
- validation (to evaluate trained ANN on known inputs it hasn't seen)
- test (for final evaluation)

until trained
  for each example in training data
    compute outputs (send example through ANN)
    compute data error (difference between ANN output and correct output)
    adjust weights to reduce error (calculus)

loss function measures difference between ANN output and correct output



overfitting can lead to learning function

if x = 2 then return 1.1
else if x = 3.5 then return 1.75
else if x = 5 then return 2.4
⋮

0 error on examples!
but no generalization

loss (error) function

categorical output		mse prefers
expected		
0.4	0.1	0.25
0.3	0.4	0.4
0.2	0.3	0.1
0.1	0.2	0.25

regression - mean squared error
multi-category classification - categorical cross entropy
binary (single class; in class or not) (is this a hotdog?) - binary cross-entropy (log loss)