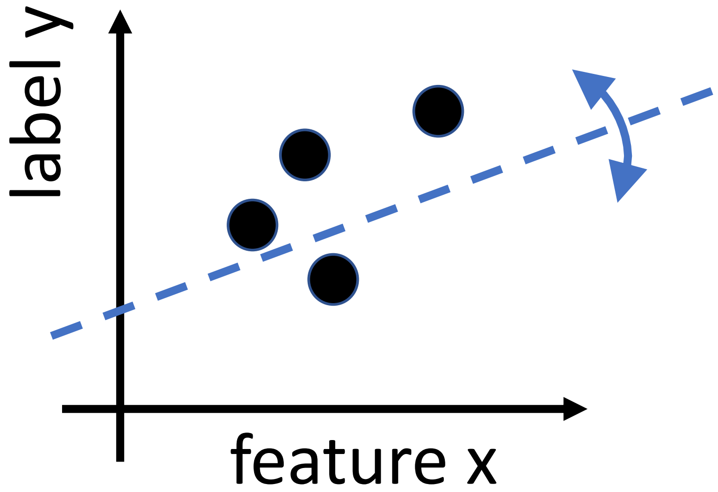
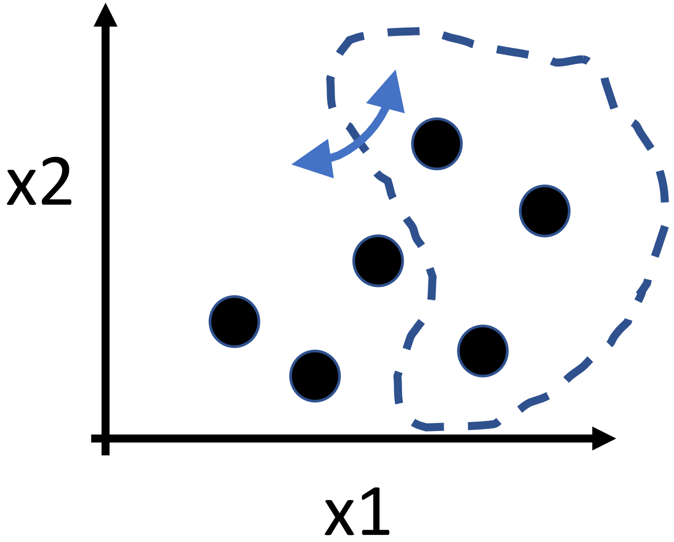
Optimization Methods in Reinforcement Learning

A. Jung, Aalto University
Helsinki, October 2020

Machine Learning is Optimization



supervised ML



unsupervised ML

reinforcement Learning

of optimal policy for steering

https://www.donkeycar.com/

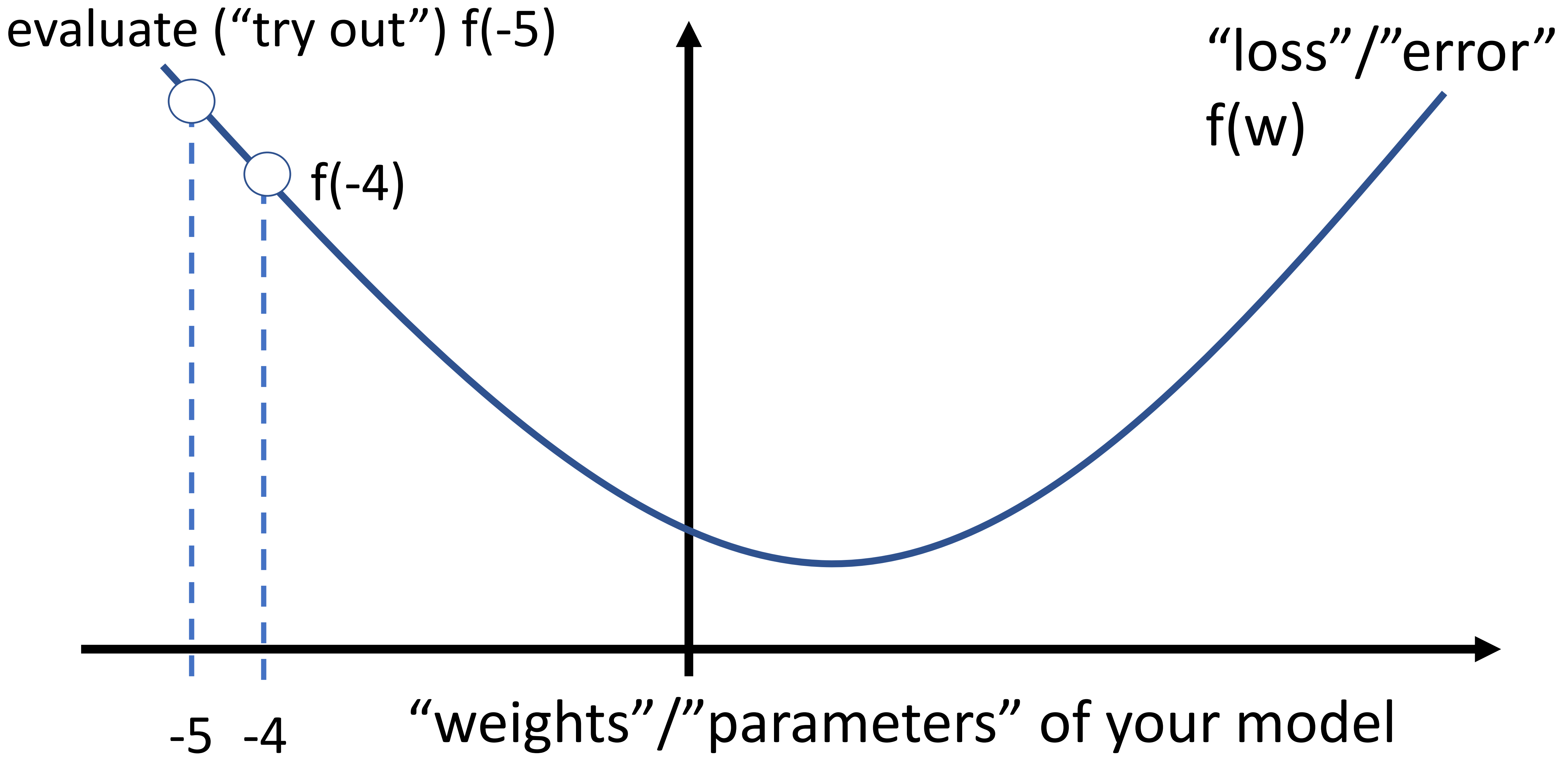
Optimization

Machine Learning

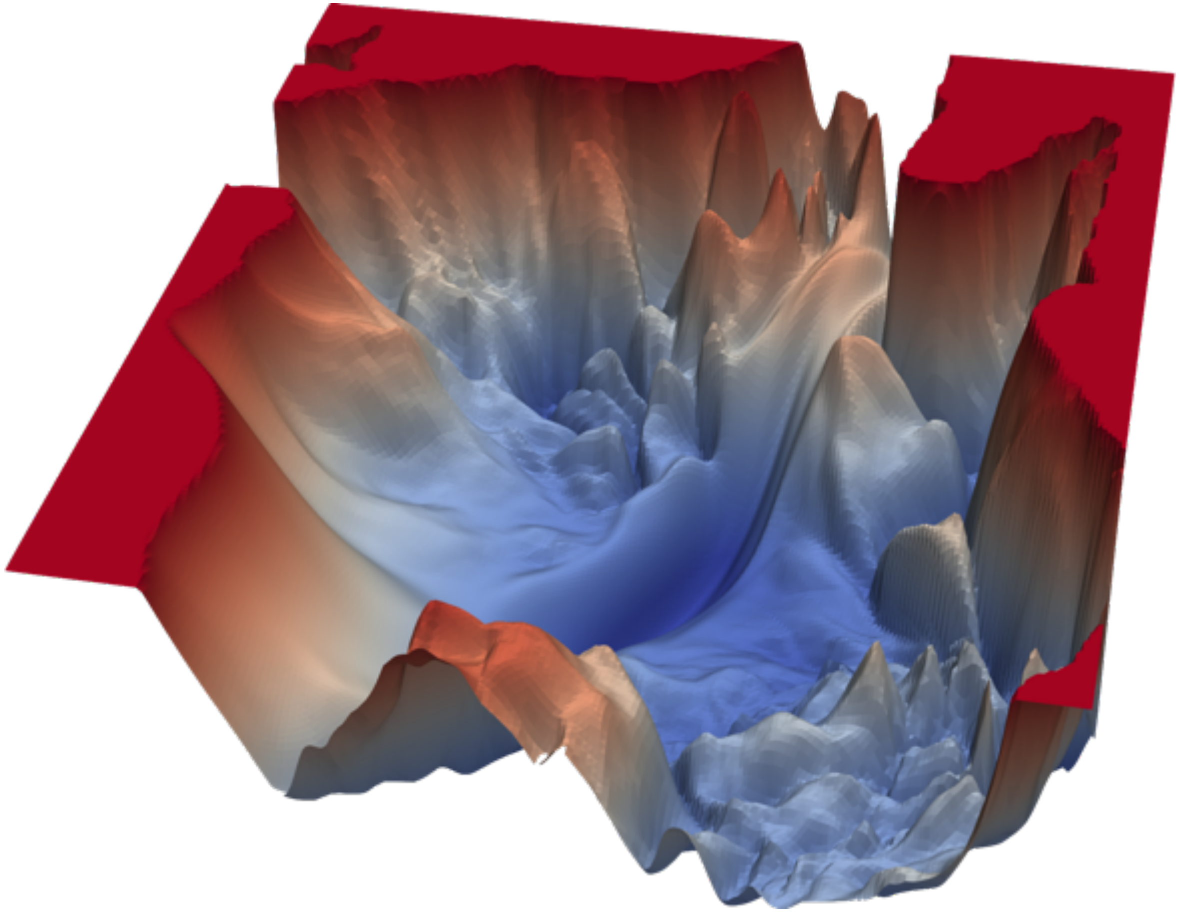
Reinforcement Learning

https://www.knoepfe.de/Zierknopf-Matroschka-Babuschka-in-Bunt

Convex Optimization



Objective/Loss Functions in Deep Learning



https://www.cs.umd.edu/~tomg/projects/landscapes/

Loss Functions in Reinforcement Learning

“try out”
weight = -4

loss = 0

“try out”
weight = -5

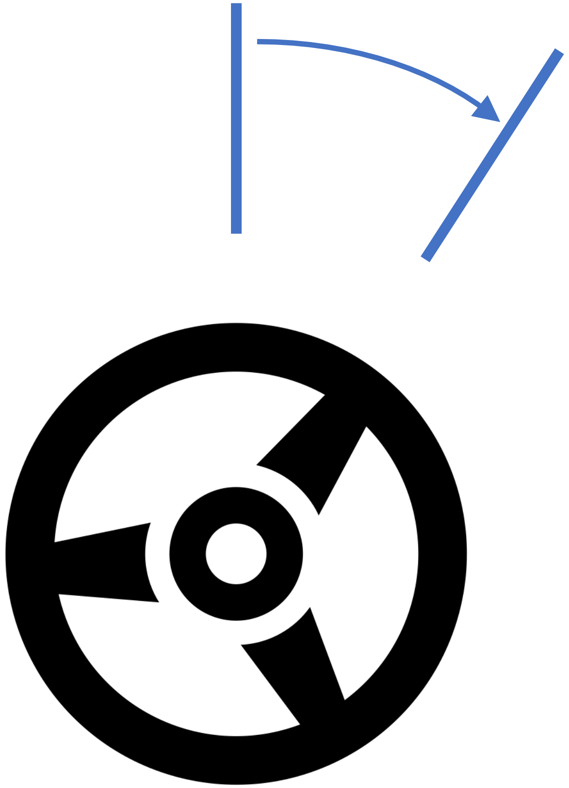
loss = 1000000

A Particular Problem

Given on-board camera snapshot, what is best steering angle?

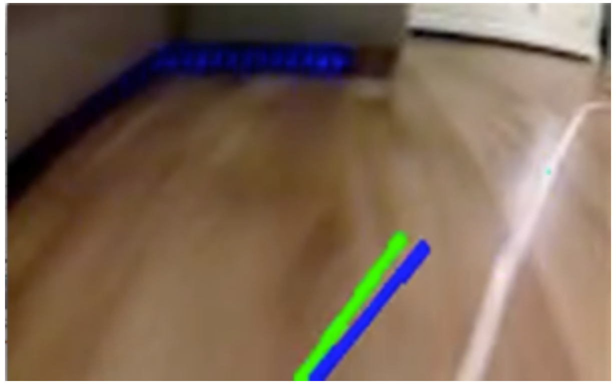
on-board camera
snapshot x

label y=30 degrees

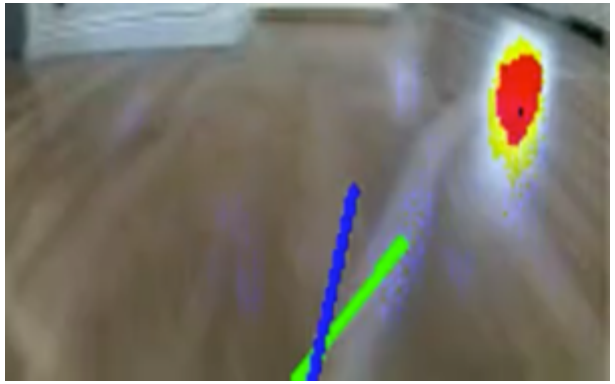


supervised learning problem: learn predictor h(x) for optimal steering angle y

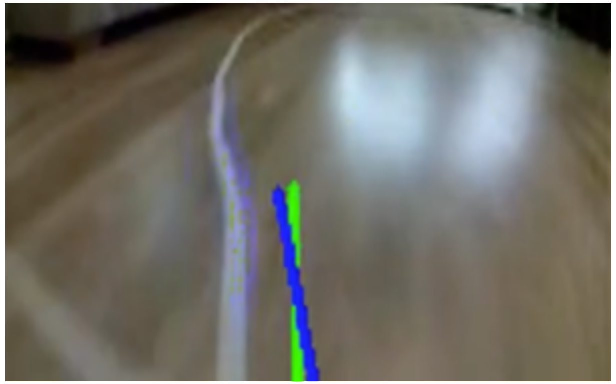
Labeled Data (by Pasi Keski-Nisula)



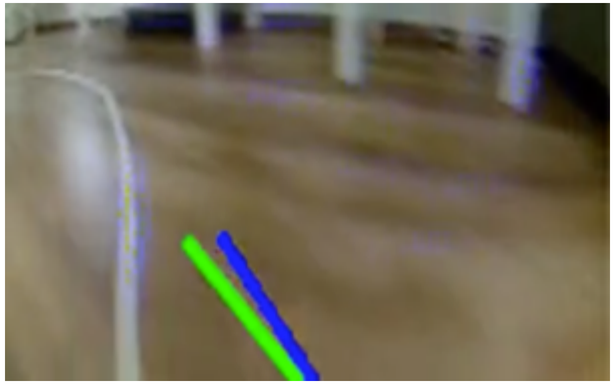
$\boldsymbol{x}^{(1)}, y^{(1)}$

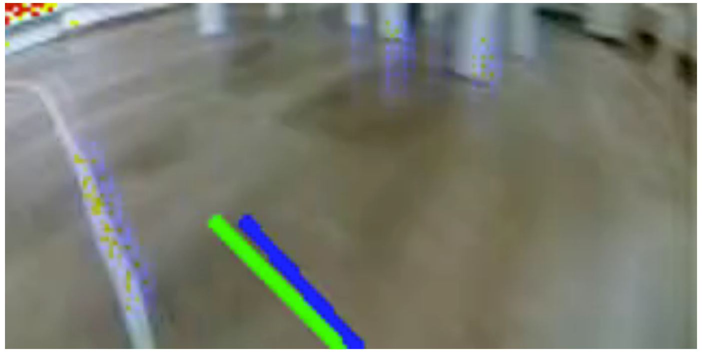


$\boldsymbol{x}^{(2)}, y^{(2)}$



$\boldsymbol{x}^{(3)}, y^{(3)}$





Learn Predictor by Min. Squared Error

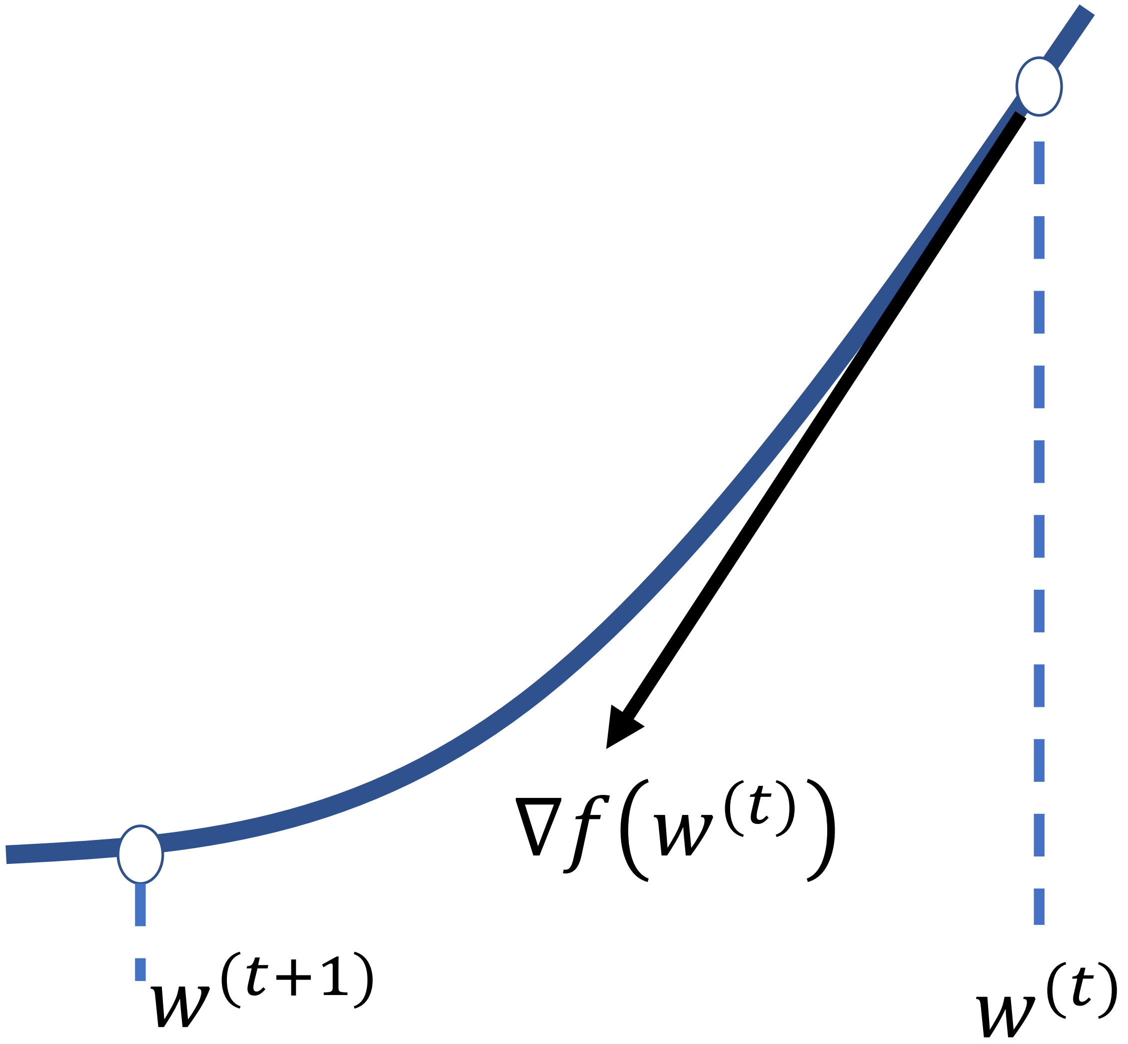
- objective function

$$f(w) = \sum_{t=1}^{T} \left(y^{(t)} - h^{(w)}\left(x^{(t)}\right) \right)^2$$

- predictor $h^{(w)}$ depends on weights w
- can probe objective function for all choices of w !!!
- gradient descent

$$w^{(t+1)} = w^{(t)} - \eta \nabla f\left(w^{(t)}\right)$$

Going Down with Gradient Descent



Online Gradient Descent

- objective function unfolds over time

$$f(w) = \sum_{t=1}^{T} f^{(t)}(w)$$

with

$$f^{(t)}(w) = \left(y^{(t)} - h^{(w)}\left(x^{(t)}\right)\right)^2$$

do gradient descent in real-time

$$w^{(t+1)} = w^{(t)} - \eta_t \nabla f^{(t)}\left(w^{(t)}\right)$$

Supervising Learning for DonkeyCar

- requires true labels y !
- optimization of fully known function
- can evaluate loss for all choices of w
- getting (accurate) labels might be difficult

Reinforcement Learning
–
Without Labeled Data!

- tune weights of steering predictor $h^{(w)}(x)$
- do not use any labeled snapshots
- only use “reward” $r^{(t)}$ as feedback
- reward might reflect if car is “on track”

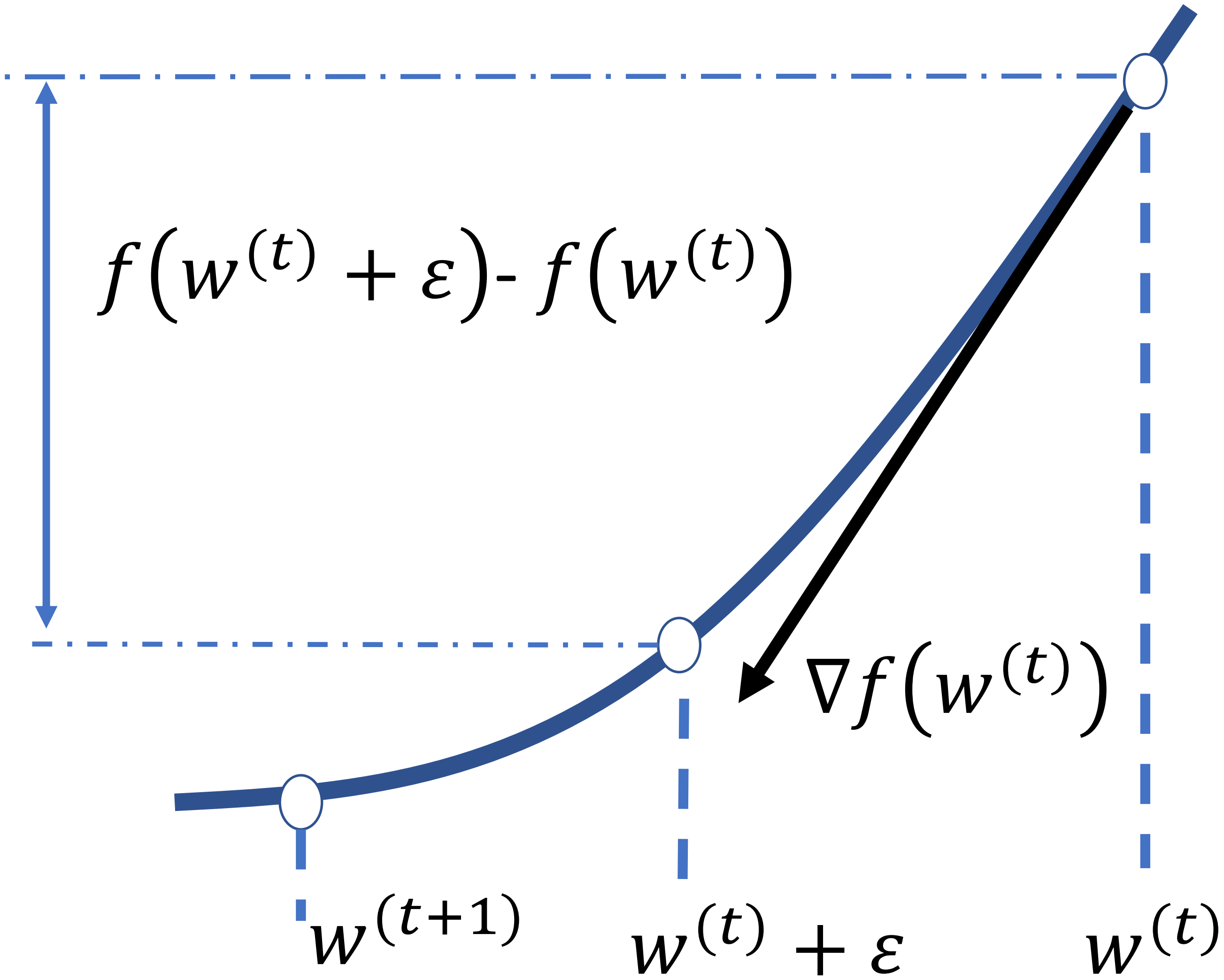
Upgrading Online Gradient Descent

- use reward as function value $f^{(t)}\left(w^{(t)}\right) = -r^{(t)}$

- cannot compute gradient since we only know few function values of $f^{(t)}$ but not entire function!

- IDEA: try out small perturbations of $w^{(t)}$ and approximate gradient with differences

Estimating Gradients by Differences



Toy Example

- two actions
  - a=1 (+5 degrees)
  - a=2 (-5 degrees)
- choose action a=1 with probability

$$P(a = 1) \triangleq \frac{1}{1 + e^{-h(x)}}$$

A Reinforcement Learning Algorithm

for each time step t:

- draw unit-norm random vector $\boldsymbol{u}$

- evaluate $h^{(w)}\left(\boldsymbol{x}^{(t)}\right)$ for $\boldsymbol{w} = \boldsymbol{w}^{(t)} + \delta\boldsymbol{u}$

- observe reward $r^{(t)}$

- gradient step $\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + \eta\, r^{(t)}\, \boldsymbol{u}/\delta$

Mirror Descent for Reinforcement Learning

- simple GD uses local linear approximations of objective
- linear approximations only based on current weights
- information in earlier iterations is “forgotten”
- mirror descent adds “regularization” to GD
- variants of MD differ in precise choice for regularizer

online GD

$w^{(t+1)} = w^{(t)} - \eta_t g^{(t)}$ with $\nabla f^{(t)}\left(w^{(t)}\right)$

can be rewritten as

$$w^{(t+1)} = \underset{w}{\operatorname{argmin}} \sum_{r=0}^{t} w^T g^{(r)} + R(w)$$

with regularization function $R(w) = \|w\|^2$

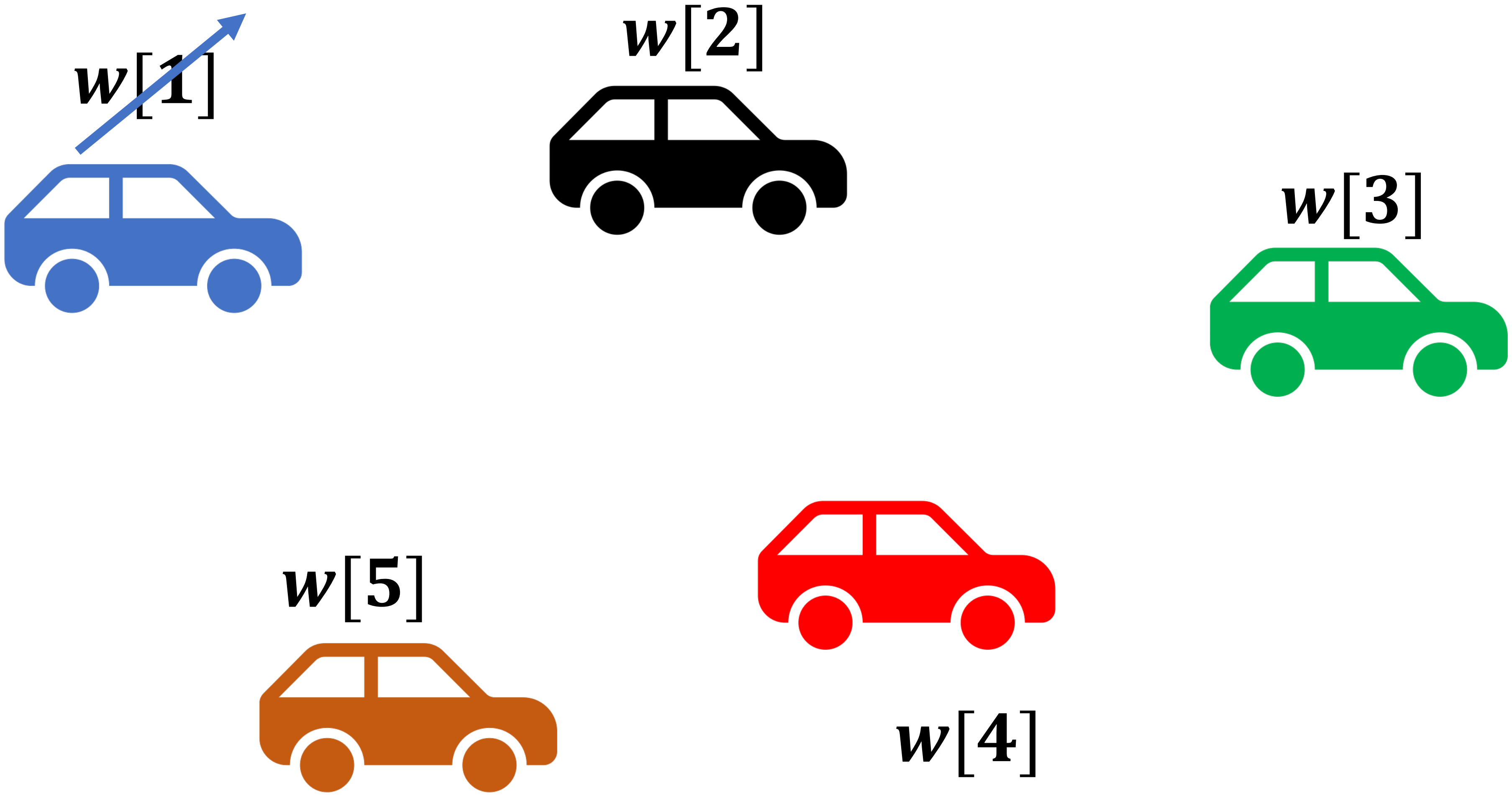
different MD algorithms obtained by different R

MD Optimal for MAB

Tsallis-INF: An Optimal Algorithm for Stochastic and Adversarial Bandits

Julian Zimmert ZIMMERT@DI.KU.DK
Yevgeny Seldin SELDIN@DI.KU.DK
University of Copenhagen, Copenhagen, Denmark

MD for Multi—Agent RL



MD for Multi—Agent RL

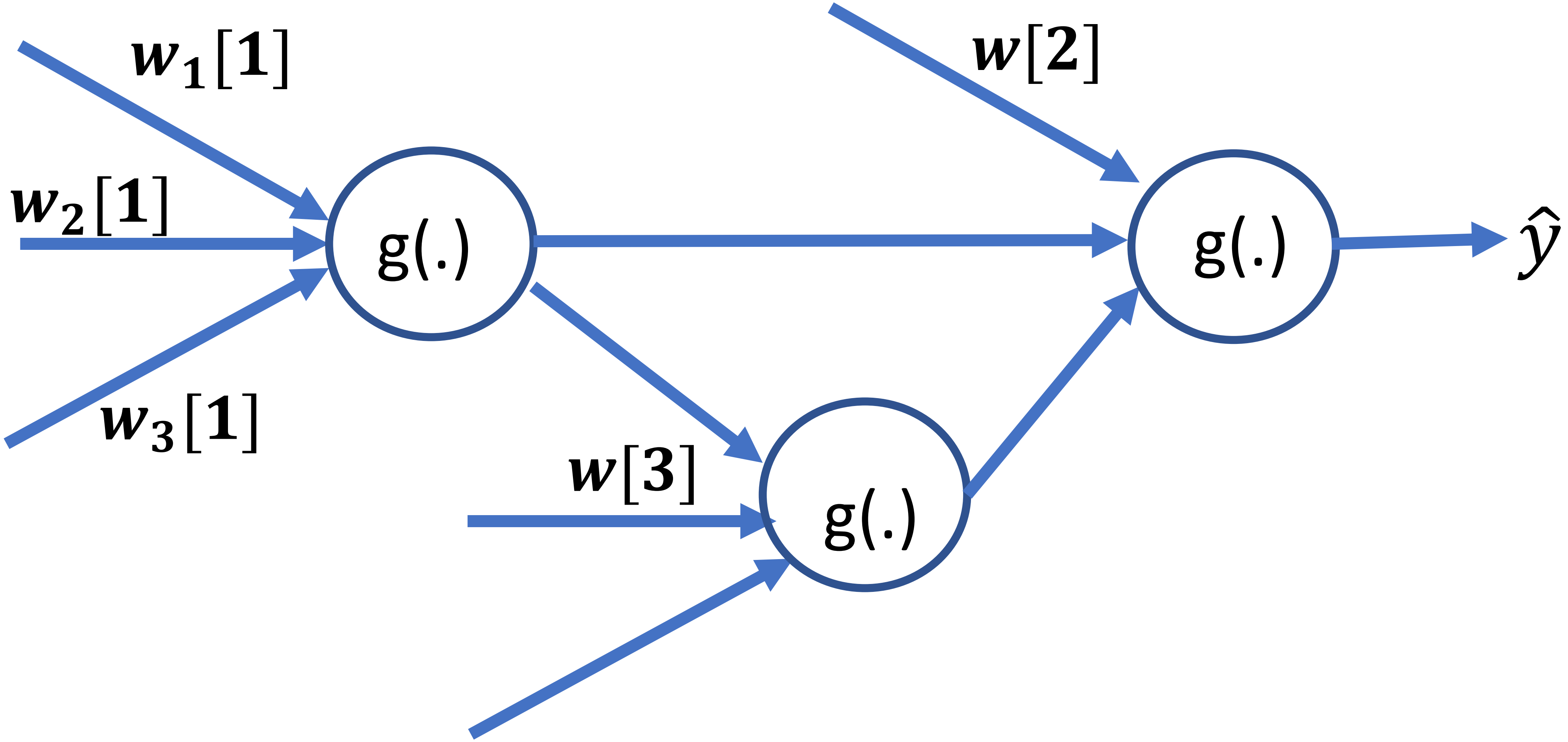
Learning in games with continuous action spaces and unknown payoff functions

Panayotis Mertikopoulos, Zhengyuan Zhou

▶ To cite this version:

> Panayotis Mertikopoulos, Zhengyuan Zhou. Learning in games with continuous action spaces and unknown payoff functions. Mathematical Programming, Series A, Springer, 2019, 173 (1-2), pp.465-507. 10.1007/s10107-018-1254-8 . hal-01382282

Deep Learning as Multi-Agent RL



Final Slide

RL=optimize unknown objective function

need to estimate gradients of objective

RL algorithms obtained by GD variants