CMSC 438/691

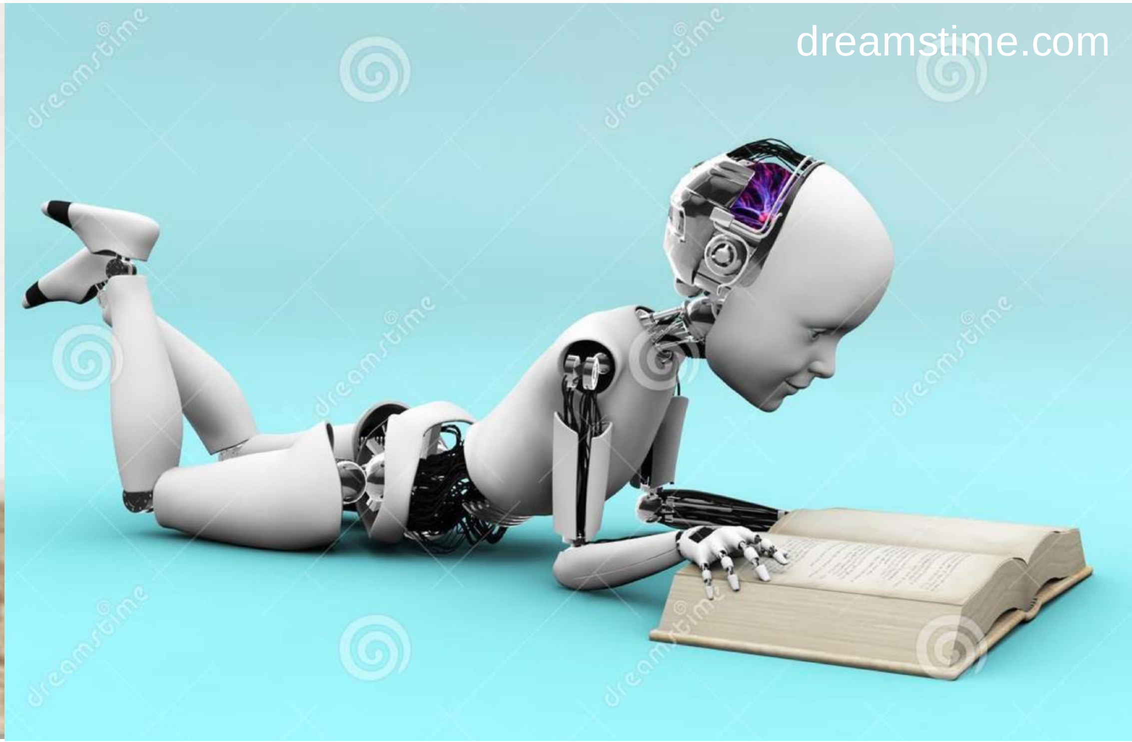
Introduction to Machine Learning

Instructor:

Dr. Tom Arodz

What kind of learning?

Learning by trial-and-error
vs. learning from knowledge sources

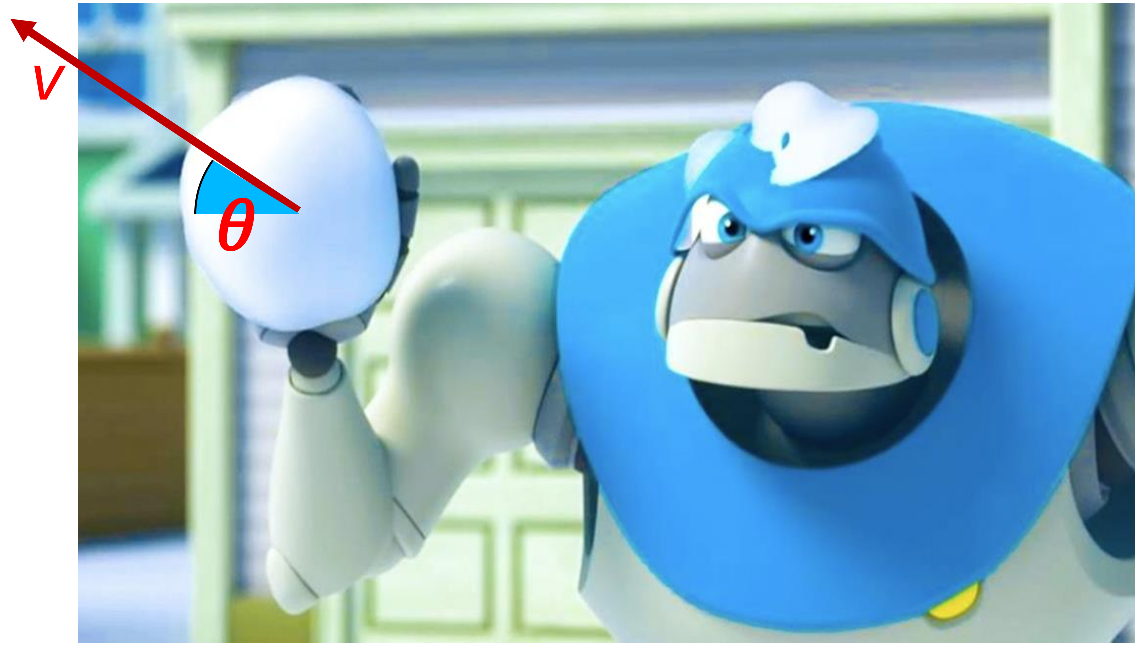


Human trial-and-error learning

Repeat:

- Throw a ball
- Observe how close to the target it hit
- Adjust the movements

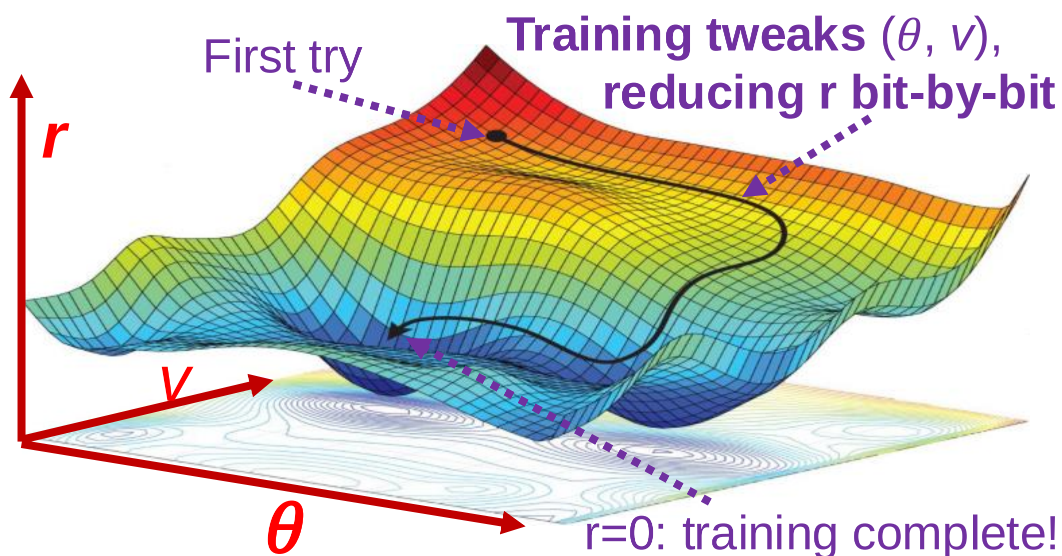
Machine learning?



Repeat:

- Throw a ball
  at angle θ
  and velocity v

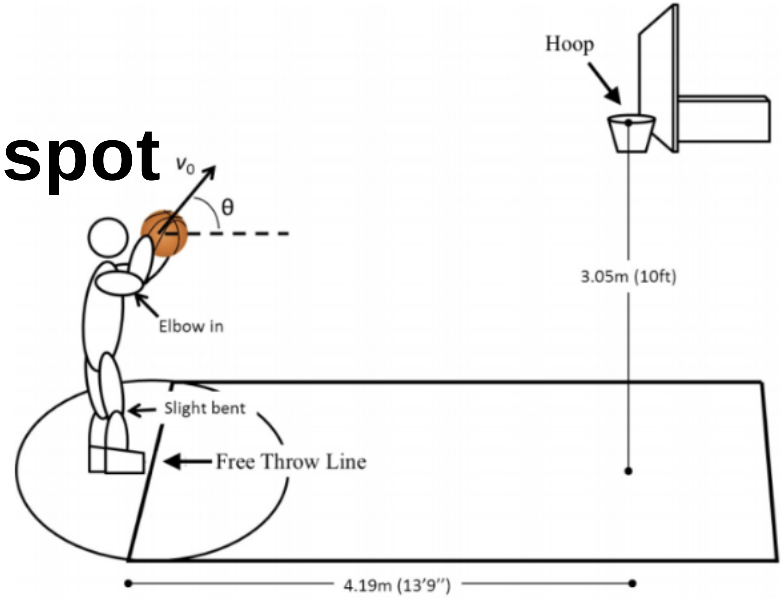
- Observe distance
  r from
  target's center
- Adjust parameters
  (θ, v) in a way
  that reduces r



Just optimization, not really ML

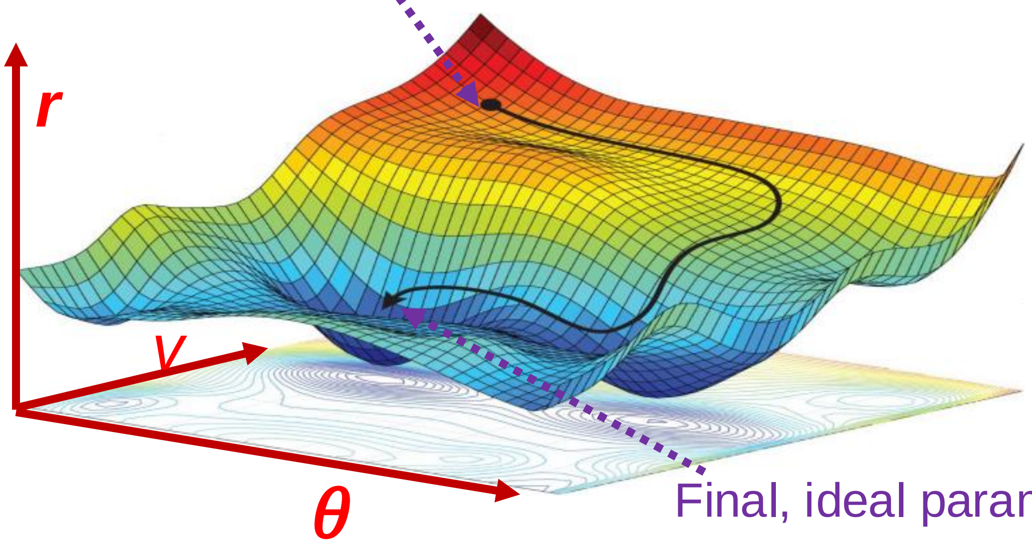
This example assumes the target is fixed in the same spot (e.g., “free throw”):

There are specific, perfect *angle* & *velocity*, eventually the optimization process will figure out what they are

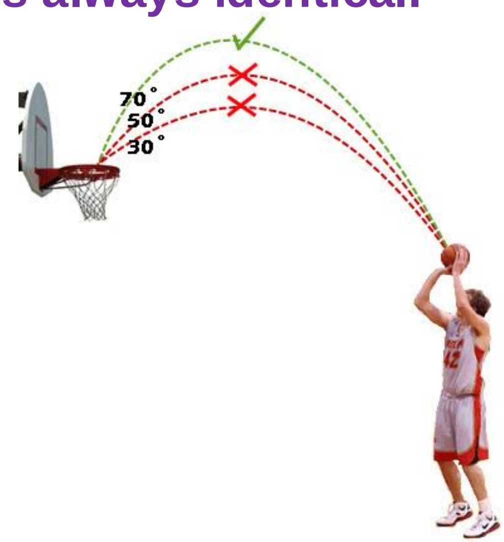


There is no varying input, the situation is always identical.

Initial, random parameters



Final, ideal parameters

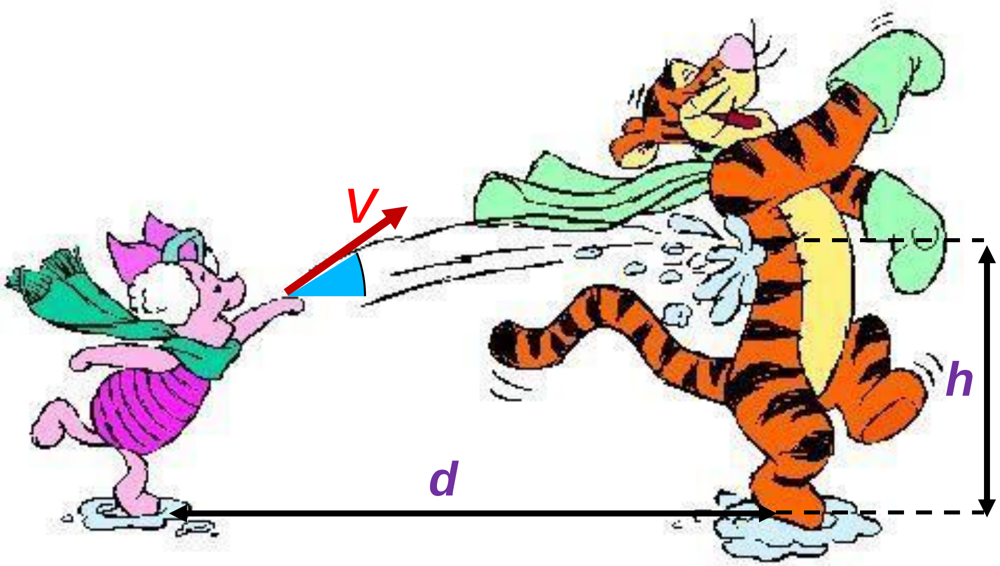


Machine learning

More often, we have some input that varies:

distance *d* of the tiger relative to the piglet,

height *h* where piglet wants the ball to land on tiger's back

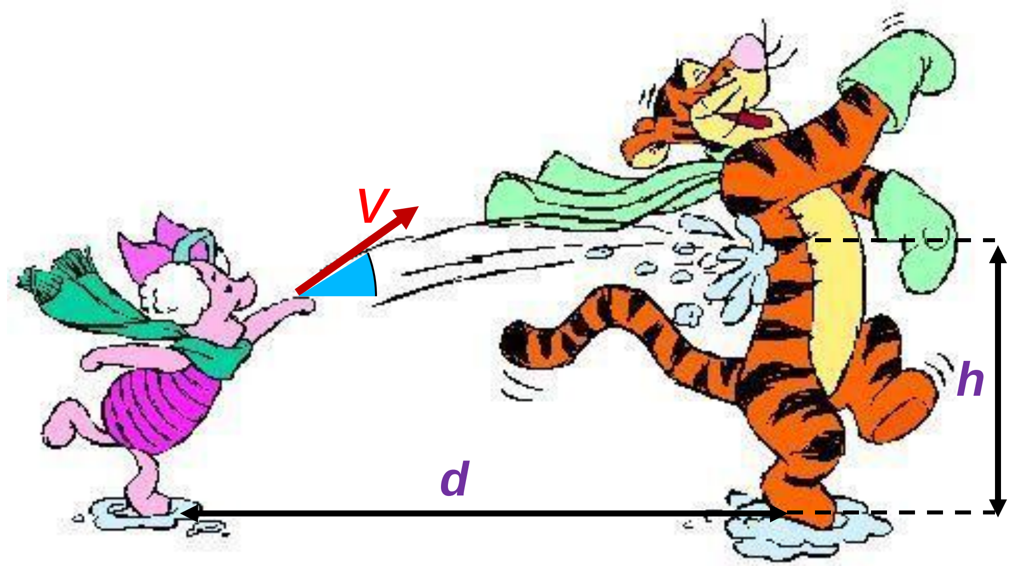


different *d, h*
mean different ideal angle & velocity

Machine learning

We construct an ML model that:

- **on input takes distance *d* and height *h*,**
- **on output produces angle θ and velocity *v***



We evaluate the model based on how far ("*r*") from the intended target the ball lands

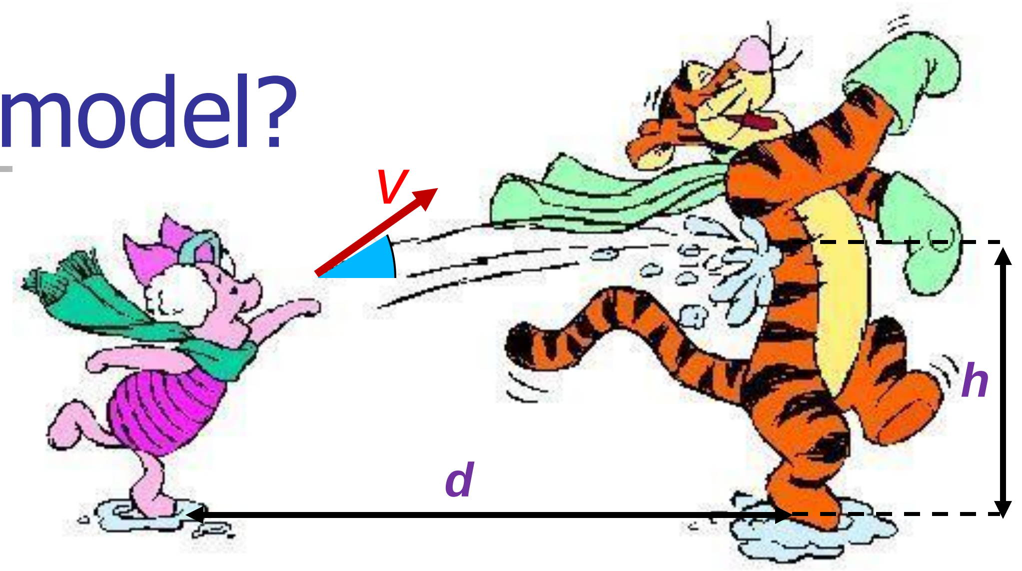
$$(d, h) \xrightarrow{\text{ML model}} (\theta, v) \xrightarrow{\text{quality metric}} r$$

ML model

quality metric

now, (θ, v) are not parameters of the model but instead its outputs

What is an ML model?



ML model is a math function that produces output from input

$(d, h) \longrightarrow (\theta, v)$

ML model

$$\left.\begin{aligned}\theta &= u*d + w*h \\ v &= s*d + t*h\end{aligned}\right\} \longleftarrow$$

ML model:
2 inputs
2 outputs
4 parameters

Key for ML: the model has some parameters that will be tweaked during training

A very simple but not unrealistic model:

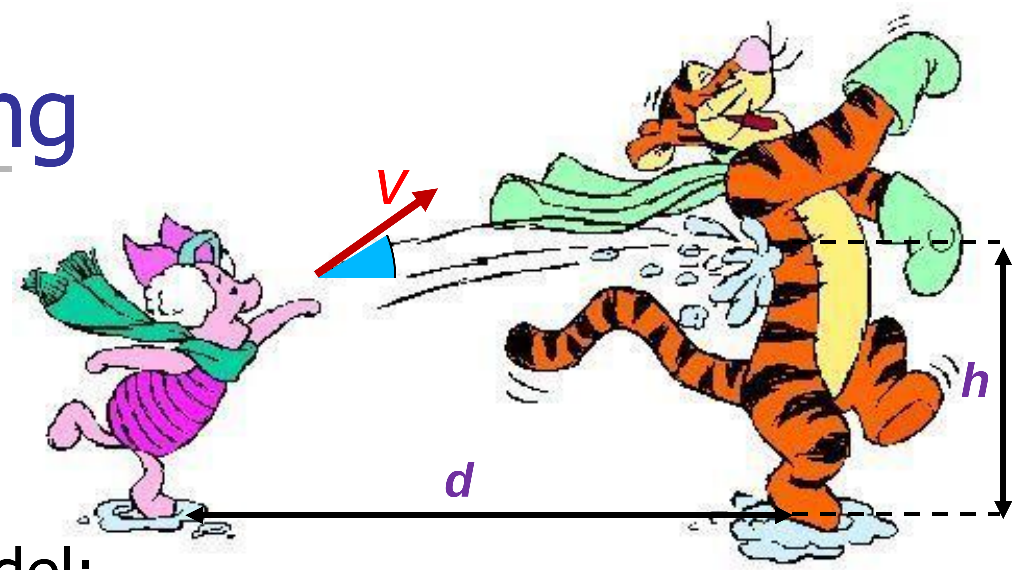
Velocity *v* should be higher if distance to tiger is larger (and maybe also if target is higher on his back).
How much higher should *v* be?
That's specified by parameters s, t.

Angle θ should be larger if target higher on tiger's back (and maybe also if tiger is farther).
How much larger should θ be?
That's specified by parameters u, w.

Machine learning

Repeat:

Take inputs *d*, *h*



Calculate outputs of ML model:

$\theta = u*d + w*h$

$v = s*d + t*h$

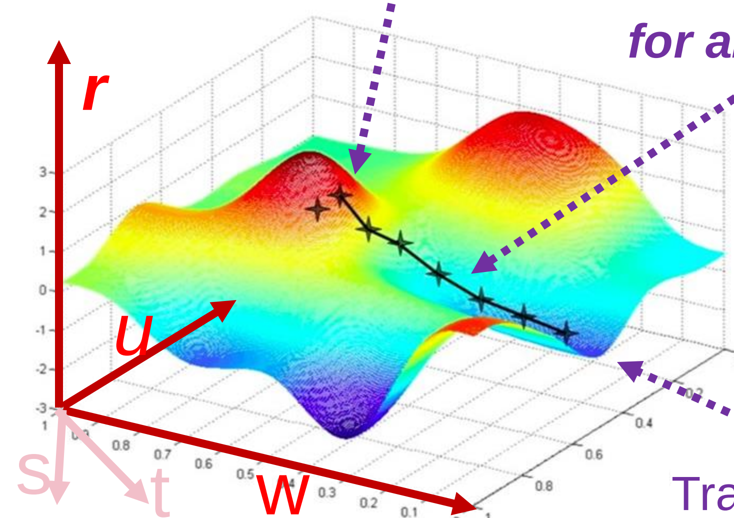
ML model

ML model parameters

Training tweaks (*u, w, s, t*), **leading to low *r*** ***for any d, h***

Calculate quality metric: distance *r* from target's center

First try



Adjust parameters (*u, w, s, t*) in a way that reduces *r*

Training complete

Machine learning

ML model:

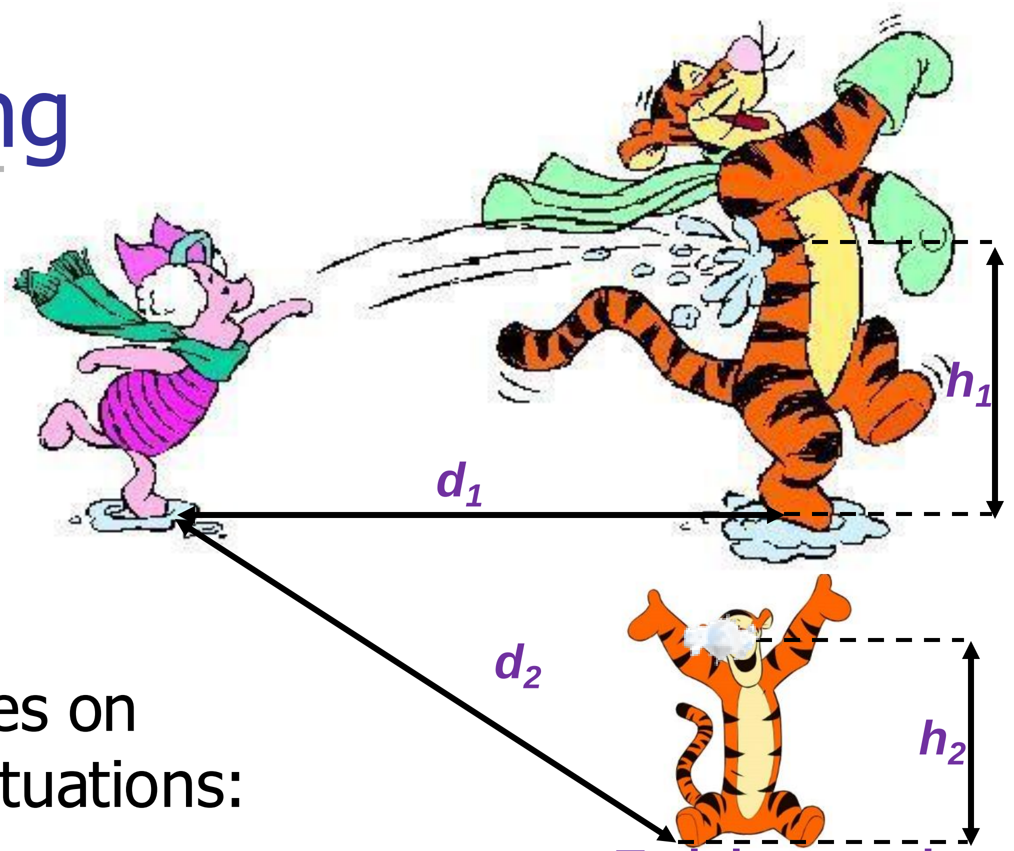
$$\begin{bmatrix} \theta \\ v \end{bmatrix} = \begin{bmatrix} u & w \\ s & t \end{bmatrix} \text{x} \begin{bmatrix} d \\ h \end{bmatrix}$$

Quality metric:

$r(\theta, v) = r(d,h\,;\,u,w,s,t\,)$

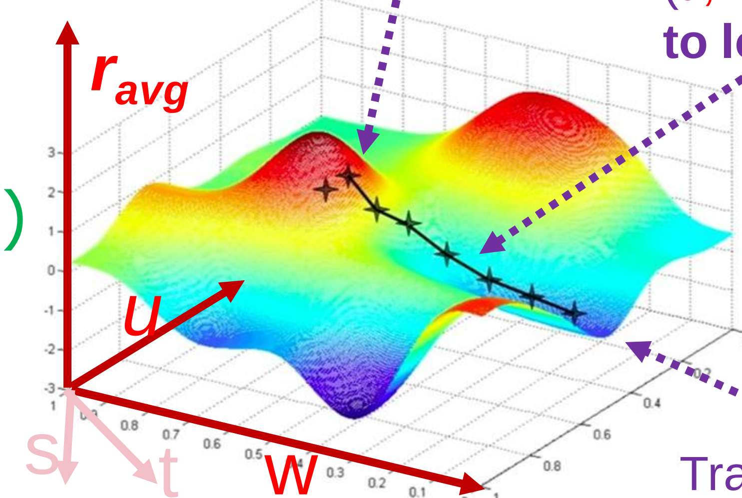
The optimization actually focuses on average quality over many situations:

$r_{avg} = \Sigma_i\, r(d_i, h_i\,;\,u,w,s,t\,)$



A key challenge in ML:

good performance (low r_{avg}) on situations seen during training may not translate to good performance later



Machine learning

Repeat:

Take inputs *d*, *h*

Calculate outputs of ML model:

$$\begin{bmatrix} \theta \\ v \end{bmatrix} = \begin{bmatrix} u & w \\ s & t \end{bmatrix} \times \begin{bmatrix} d \\ h \end{bmatrix}$$

Calculate quality metric:
distance *r* from
target's center

Adjust parameters
(*u*, *w*, *s*, *t*)
in a way that reduces r_{avg}

Course outline:

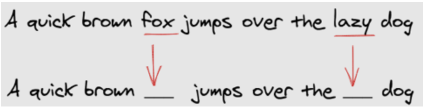
Architectures of ML models:
linear models,
deep models,
specialized models

Suitable quality metrics
(i.e., loss functions)

Training algorithms
(typically using partial
derivatives, e.g. ∂r/∂u)

Machine learning

This approach is very generic:

Domain		**Inputs**	**Outputs**
Image recognition		pixels	probabilities of 0/1
Image generation		pixels	pixels
Human language		words (some masked)	probabilities of each possible word
Medical diagnosis		vital signs, gene expression, etc.	probability of healthy / sick

We just need to pick a proper model architecture that produces output from input, and has trainable parameters. Then we present (lots) of input-output data pairs, and train the model (tweak parameters) to minimize some quality metric (avg. over in-out pairs).

Topics covered in the course

- Introduction to machine learning
  - Probabilistic view of machine learning, Bayes classifier
  - Loss, risk, empirical risk minimization
  - Gradient descent and automated differentiation
- Linear models
  - Logistic regression
  - Support vector machines
- Deep nonlinear models
  - Fully connected neural networks
  - Convolutional networks
  - Generative networks (incl. diffusion models)
  - Large language models

Lecture topics (tentative, from last year)

- Course Intro (today)
- ML Example: Apples vs Oranges
- ML Assumptions / When to use ML
- Probabilistic View of ML
- Maximum Likelihood Estimation
- Bayesian Learning
- Trainable Probabilistic Models
- Error Metrics
- Loss Functions
- Cross-Entropy
- Intro to PyTorch
- Nonlinear Kernel Methods
- Nonlinear Features and Neural Networks (Multi-Layer Perceptron)
- Intro to Tensorflow
- Details of an MLP Layer
- MLP: Universal Approximation
- MLP: Why Depth
- Training Deep MLPs
- Convolutional Neural Networks
- Adversarial Examples
- Normalizations and Skip Connections
- Weight Init, and Dropout
- Language Models: Intro
- Self-Attention and Transformers
- Large Language Models
- Modern Models for Vision Tasks (CLIP, Vision Transformer)
- Diffusion-based Generative Models

Prerequisites

- The course will assume prior background in:
  - *python (basic coding & execution ability)*
  - *general math (proofs and derivations)*
  - *linear algebra (matrix/vector),*
  - *multivariate calculus (derivative),*
  - *statistics (probability, conditional probability)*

Zoom and Canvas

- Zoom details:
  - https://vcu.zoom.us/j/82195507595
  - logging to zoom with VCU credentials required
- For convenience, the meetings will be also recorded to Canvas
  - Slides + video posted typically <24hrs after class
- Homework assignments, announcements, etc., will also be posted on Canvas

Zoom and Canvas (cont'd)

- This course is classified as:
  - CMSC 438 (undergrad.): in-person, E3229
  - CMSC 691 (grad.): online, synchronous (i.e., live Zoom)
- In practice:
  - graduate students are welcome to show up in-person (space permitting), especially if you expect to have questions/discussion
  - undergraduate students are not required to show up in person (no attendance is taken)
  - tests will be done online, from home, via Canvas
  - some of the classes (esp. in the second half of semester, where we will have less theory, and more pytorch/tensorflow codes) will be done fully online

Grading – components

- Homework projects 70%
  - 5 homework projects, 14% each
- Test 1 (basic ML theory + linear models) 15%
- Test 2 (deep models) 15%
- Undergraduate vs graduate:
  - Tests for 691 will cover more material
    (some material during class will be labeled “graduate only”)
  - Homework for 691 will be more complex

Grading – components

- Homework projects
  - To be done individually, in Python, in ~10 days
  - Submit on time. Late submission will be accepted for up to 48 hours.
    - *Late >15min and <=24hrs*: –1 point (max 13 instead of 14 pct points)
    - *Late >24hrs and <=48hrs*: –2 points (max 12 instead of 14 pct pts)
    - *Late >48hrs: not accepted*, 0 points

    In special circumstances, ask for extension in advance, before the due date.
- Test 1 & 2
  - Will be done electronically in Canvas
  - Study problems will be posted >1 week before each test

Grading scale

A [90% - 100%],

B [80% - 90%),

C [70% - 80%),

D [60%-70%),

F [0% - 60%)

How to reach me & the TA

- Email:
  - *tarodz@vcu.edu*
  - Subject line should start with:
    CMSC438 or CMSC691
- Office hours (instructor):
  - When: Wednesdays 12:15pm-2:15pm
    Where: E4252 / Zoom
- Office hours (TA):
  - Mohamed Salah Kamel, *kamelms@vcu.edu*
  - When: Mondays 2-3pm, Tuesdays 3:30-4:30pm,
    Where: E4222 (the usual TA room)/ Zoom

If these topics interest you,
see you on Monday!