Learning 101

Learning formulation and issues, regression, classification

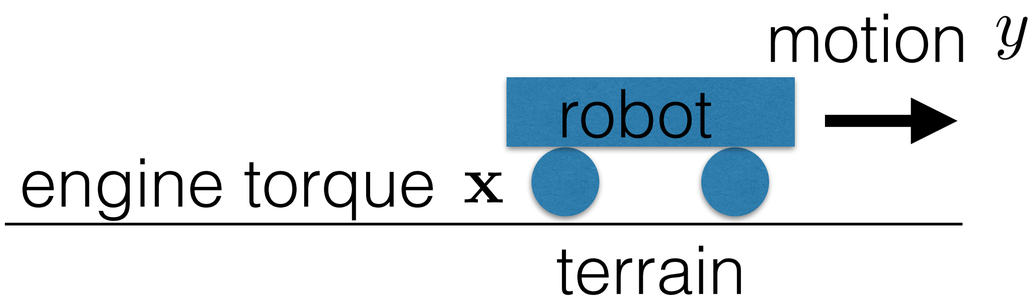
Pre-requisites:

- linear algebra,

Karel Zimmermann
Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics

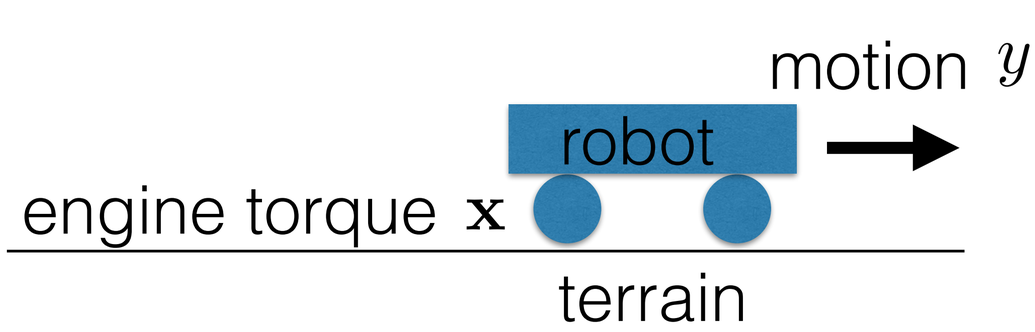
Motivation example: estimation of a motion model

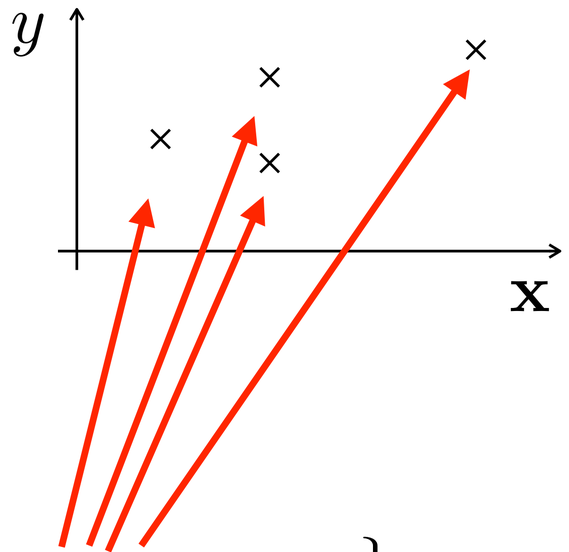
- How to create a model?
- Algorithm that maps x on y (or prob distr of y)
- This algorithm has some parameters => how to find them? => trn data



Motivation example: estimation of a motion model

- How to create a model?
- Algorithm that maps x on y (or prob distr of y)
- This algorithm has some parameters => how to find them? => loss+trn data+opt





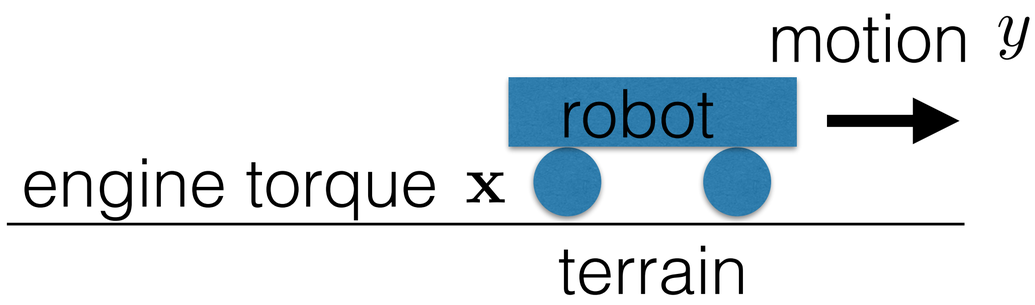
trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

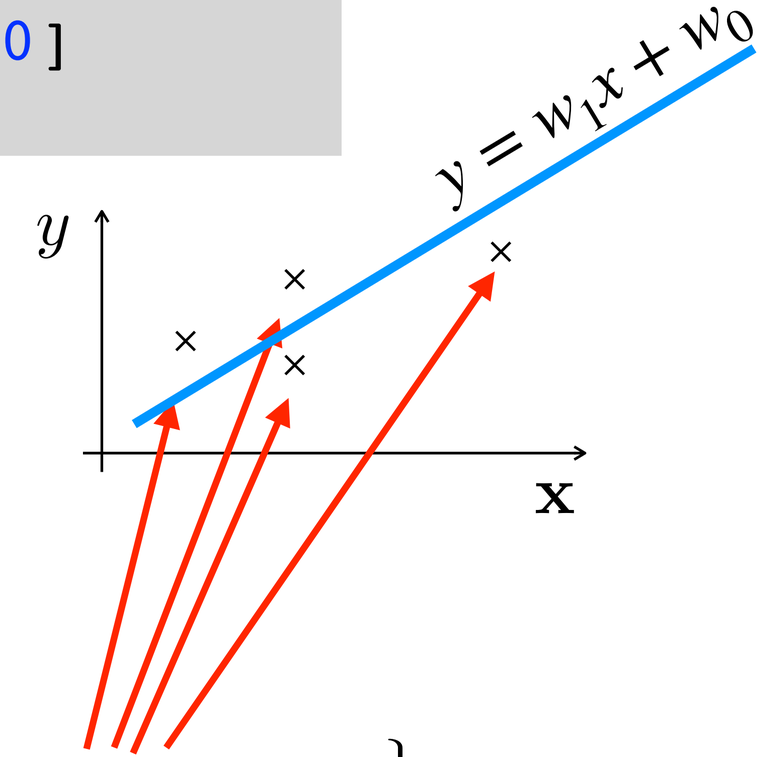
Motivation example: estimation of a motion model

- Let's implement it!

loss: $\arg\min_{\mathbf{w}} \sum_i (w_1 x_i + w_0 - y_i)^2$

```
w = torch.tensor([-2.0, 2.0], requires_grad=True)
for i in range(0, 10):
    dy = w[0] * x + w[1] - y
    loss = torch.sum(dy * dy)
    grad = torch.autograd.grad(loss, w)[0]
    w = w - 0.1 * grad
```

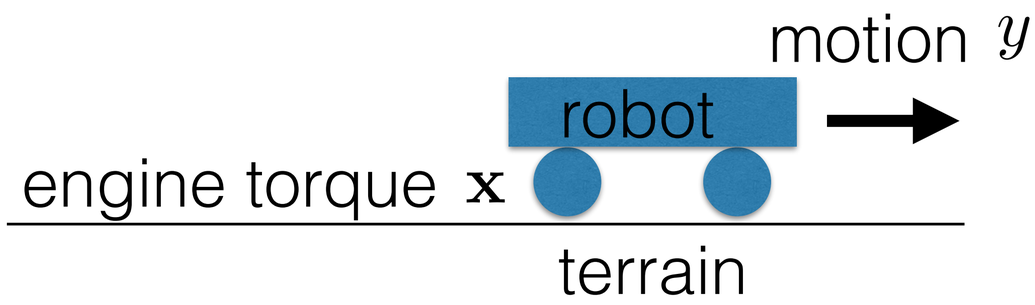


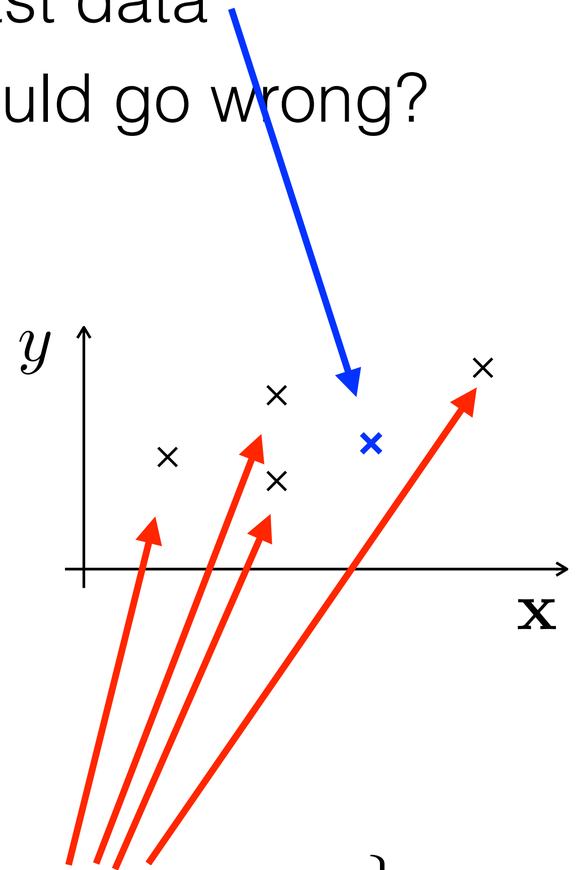


$\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

Motivation example: estimation of a motion model

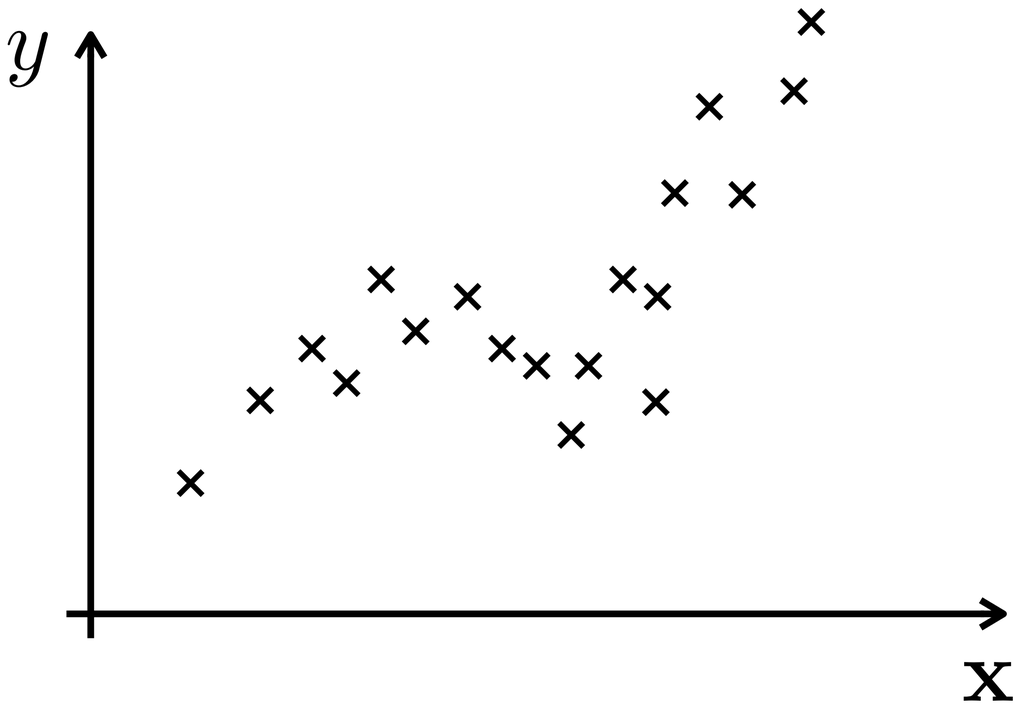
- How to create a model?
- Algorithm that maps x on y (or prob distr of y)
- This algorithm has some parameters => how to find them? => loss+trn data+opt
- How to decide that the algorithm works well? => tst data
- What if the algorithm does not work well? What could go wrong?



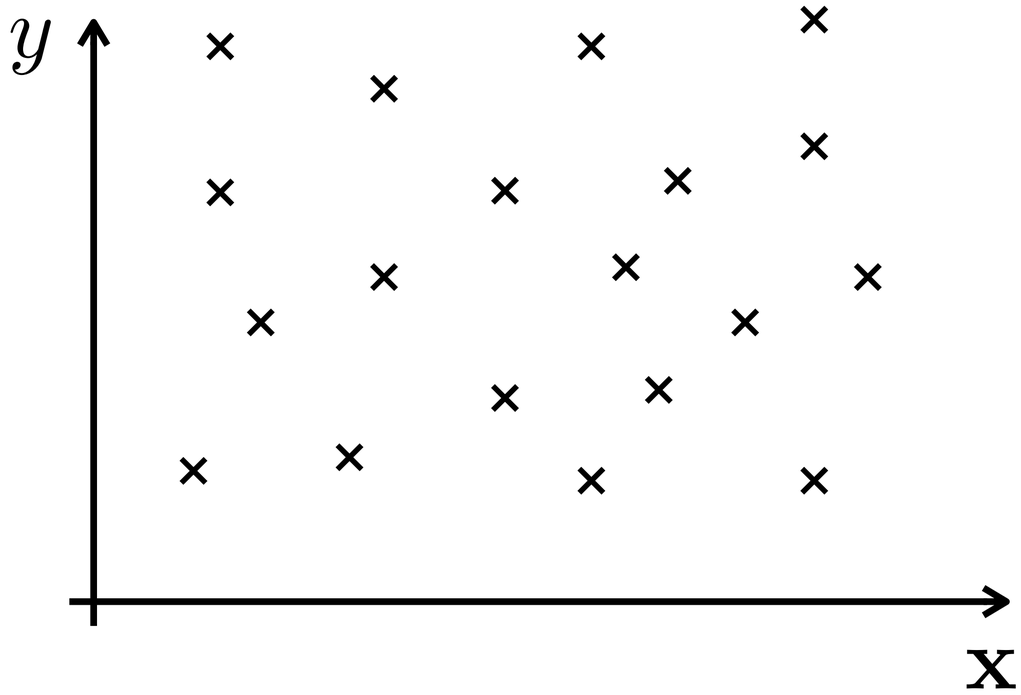


trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

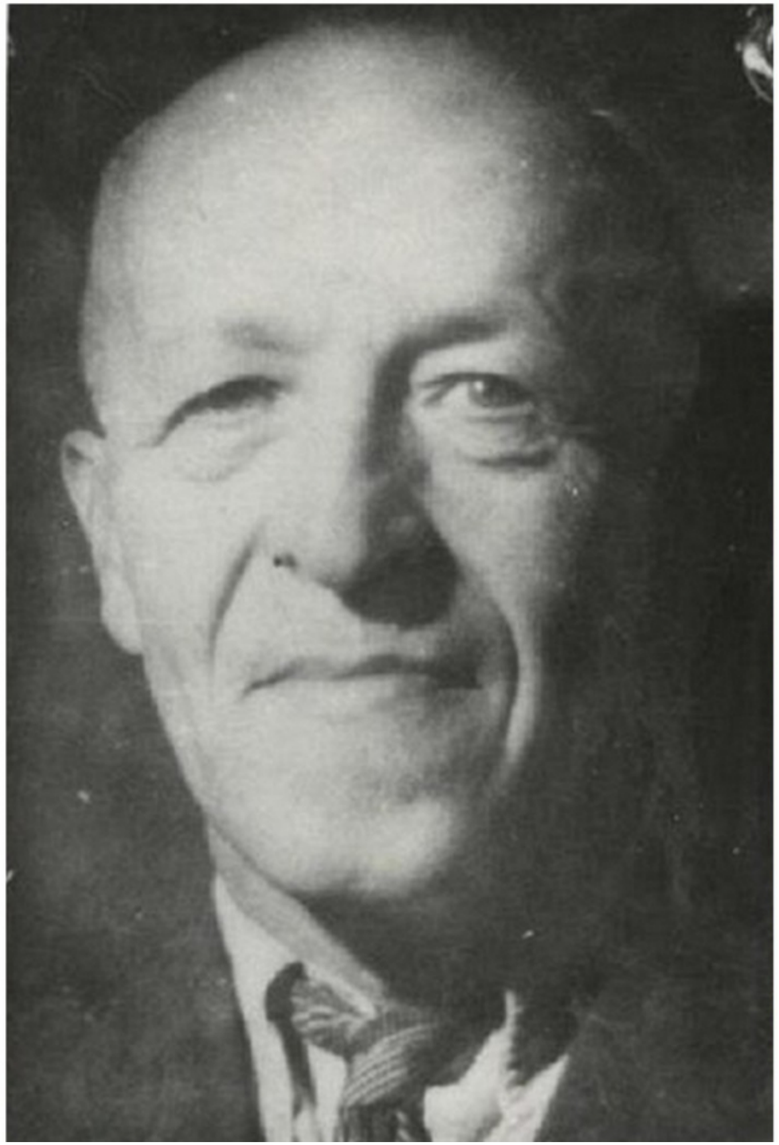
What can go wrong: **inputs x does not allow to predict y**

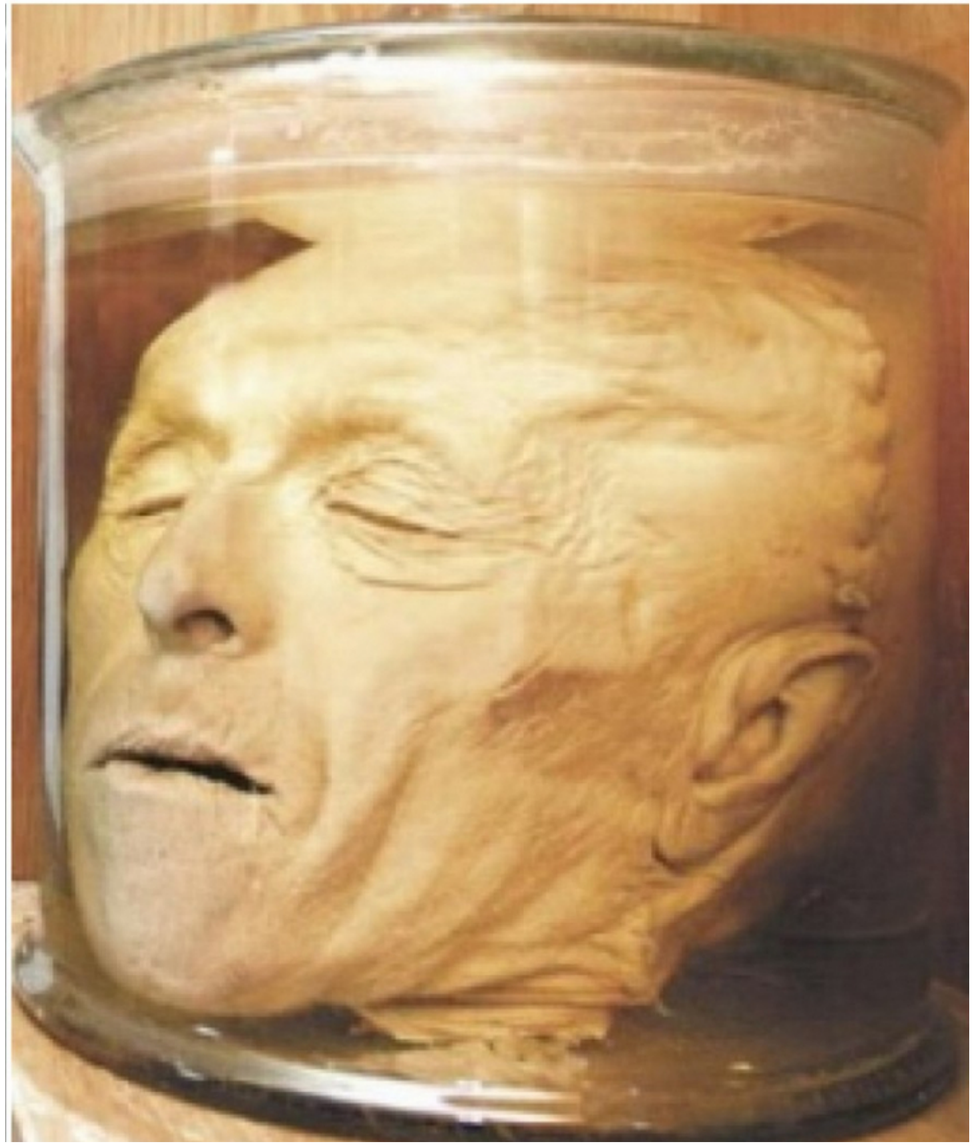


VS



What can go wrong: **inputs x does not allow to predict y**



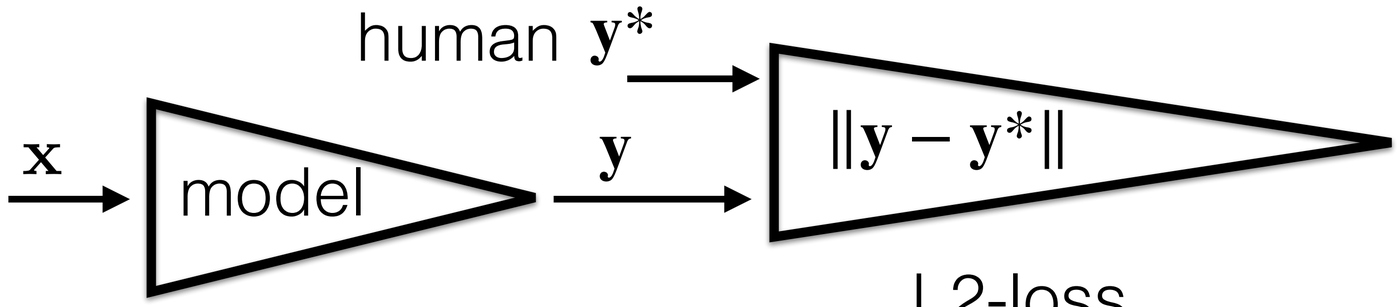


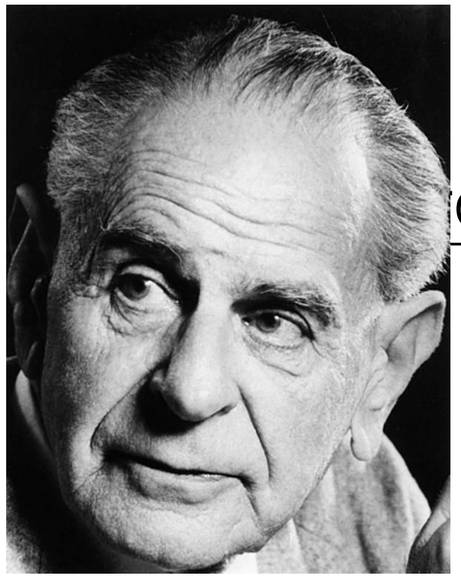
A Deep Neural Network Model to Predict Criminality Using Image Processing
https://medium.com/@CoalitionForCriticalTechnology/abolish-the-techtoprisonpipeline-9b5b14366b16

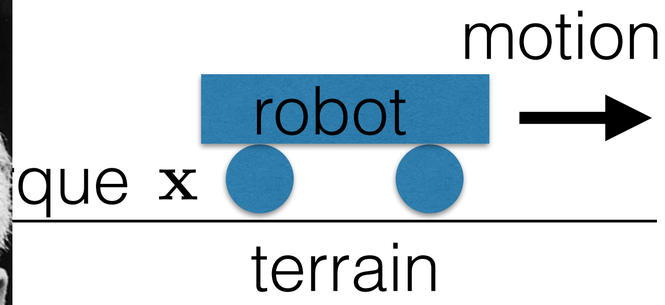
What can go wrong: **trn/tst data distribution mismatch**

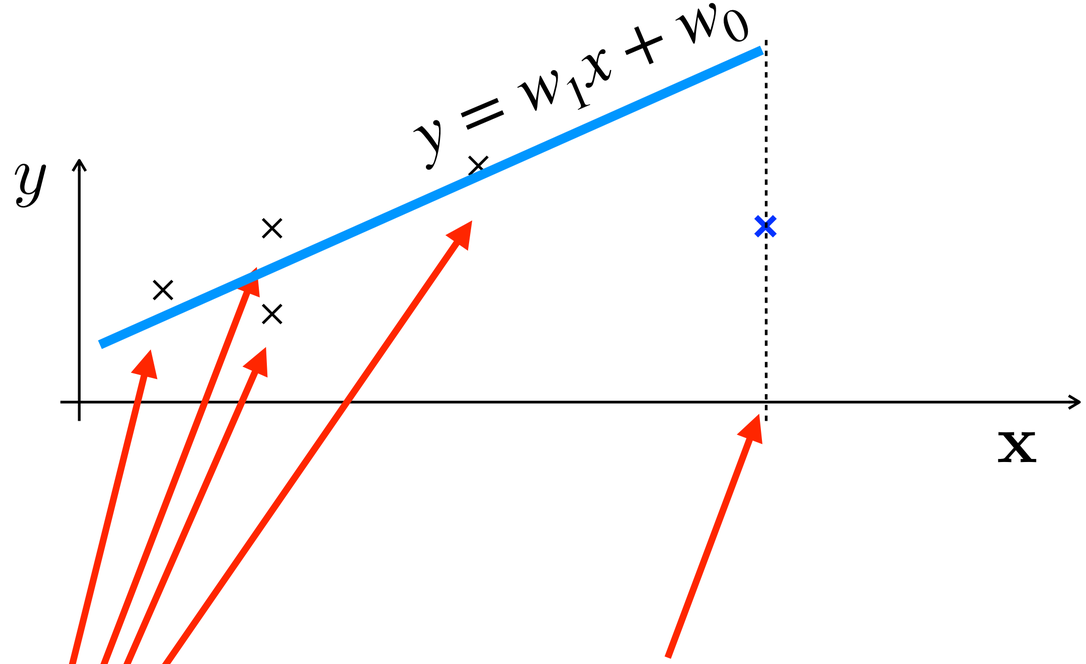
[NVidia, CVPR, 2016]

Statistical consistency issues





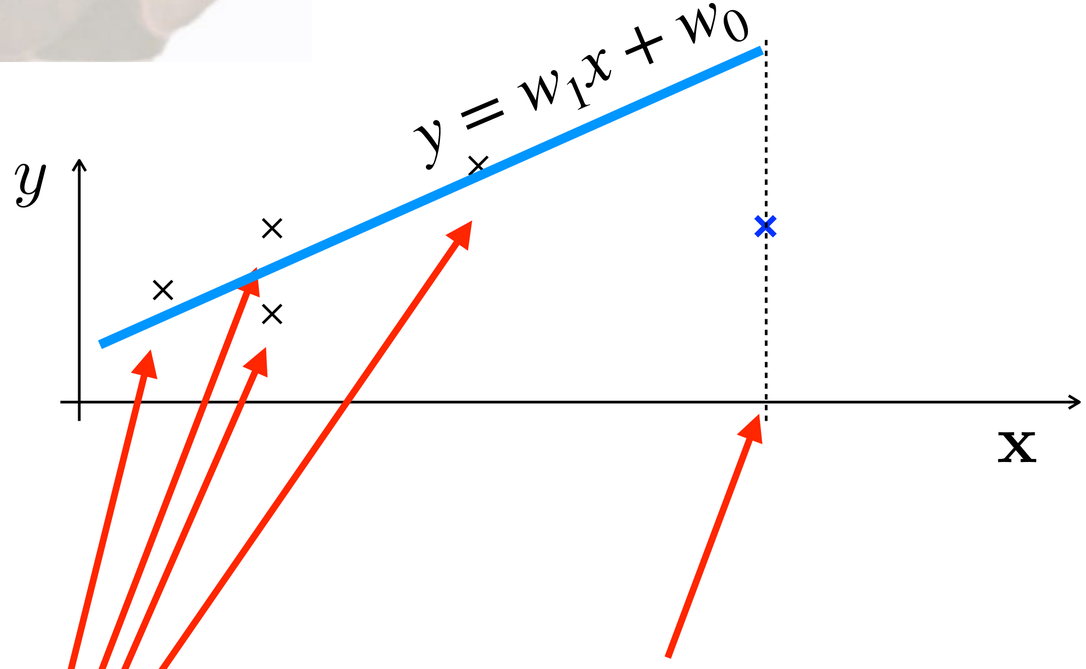




trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

tst data:

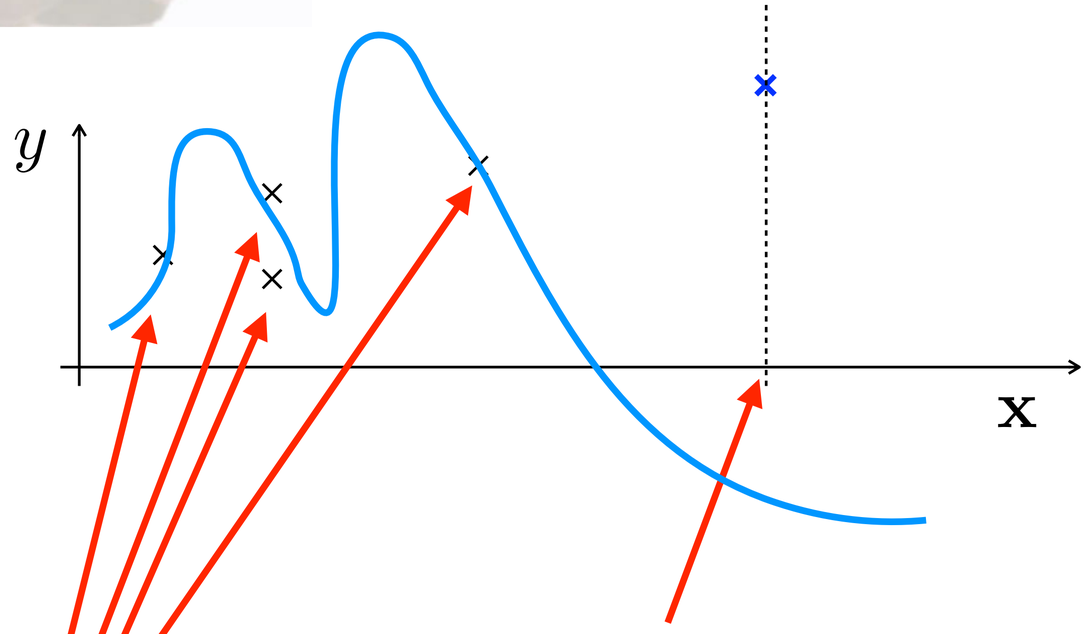
What can go wrong: **model does not generalize well**



trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

tst data:

What can go wrong: **model does not generalize well**

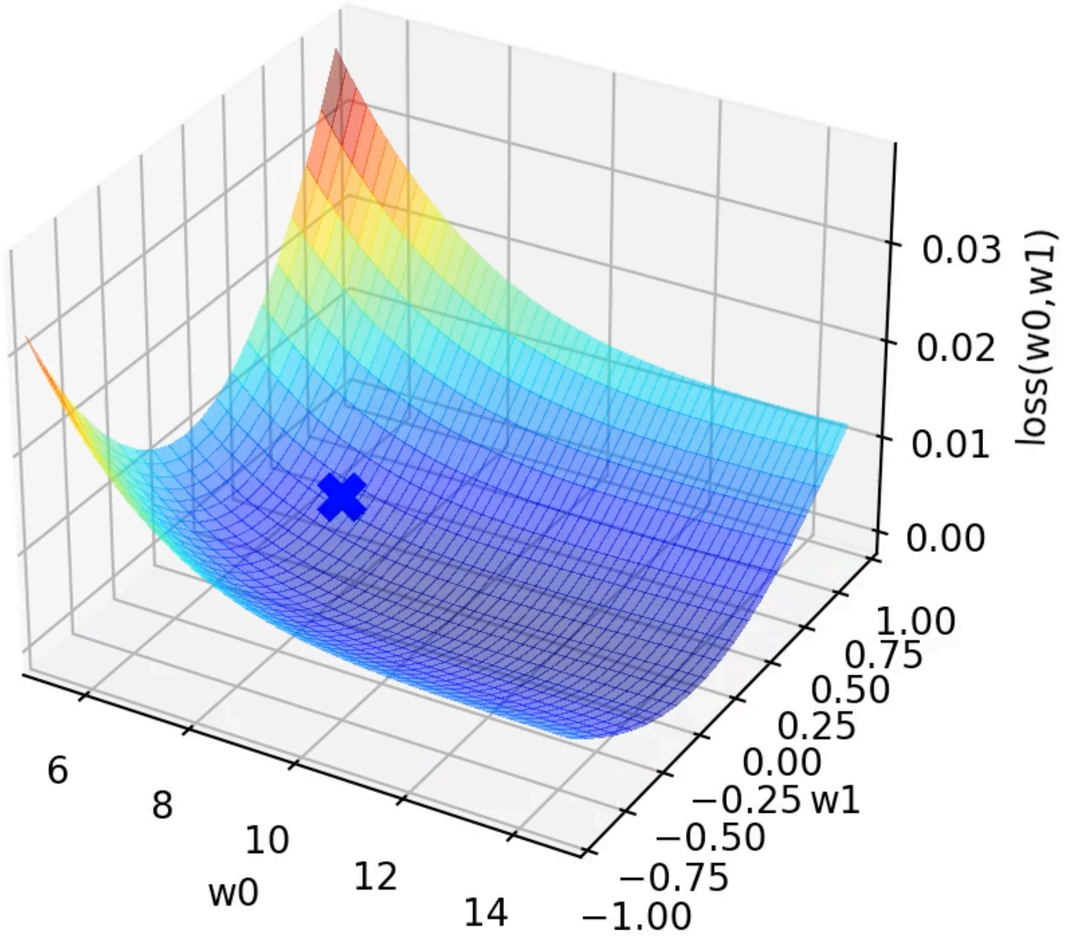


trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

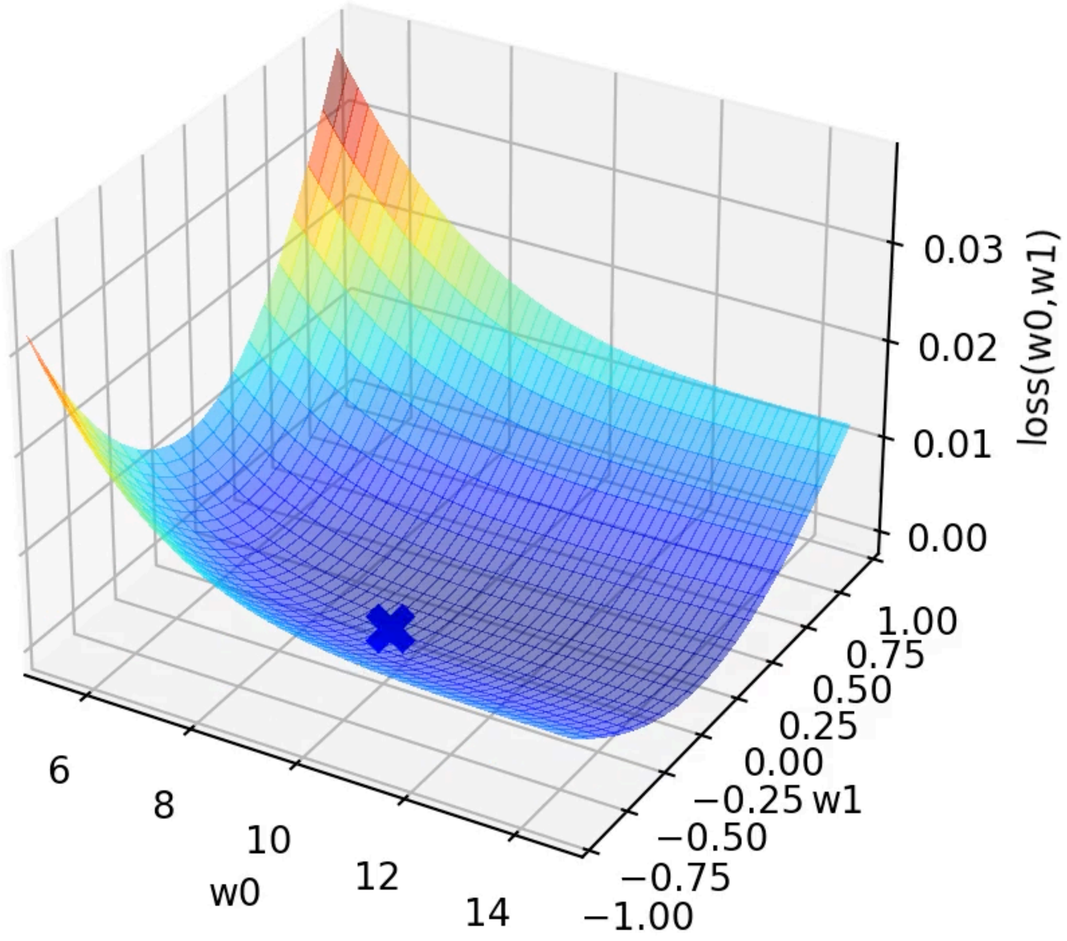
tst data:

What can go wrong: **learning fails to find good parameters**

learning_rate=30

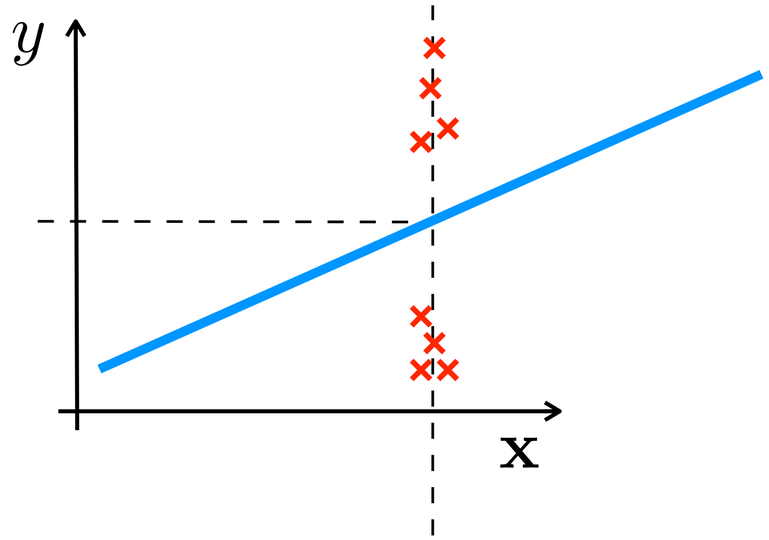


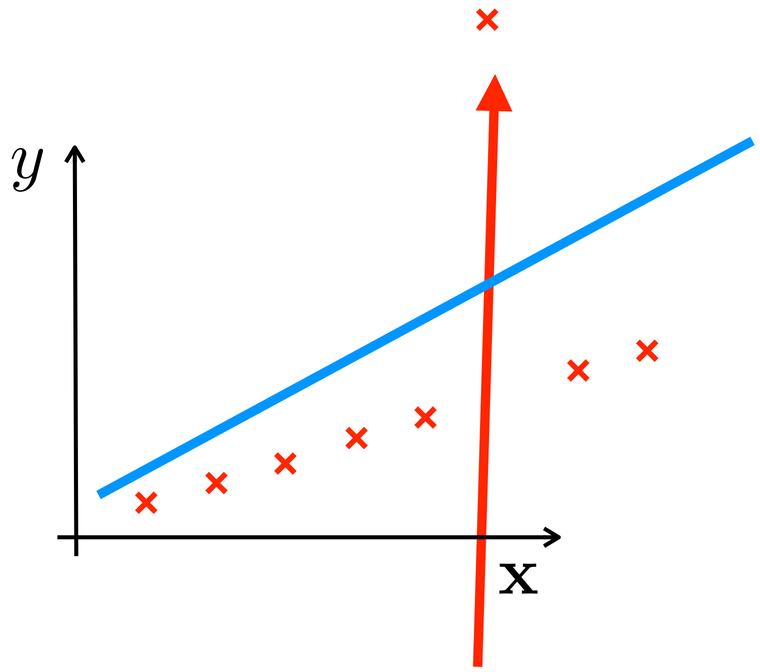
learning_rate=80



What can go wrong: **inappropriate choice of loss function**

left/right steering

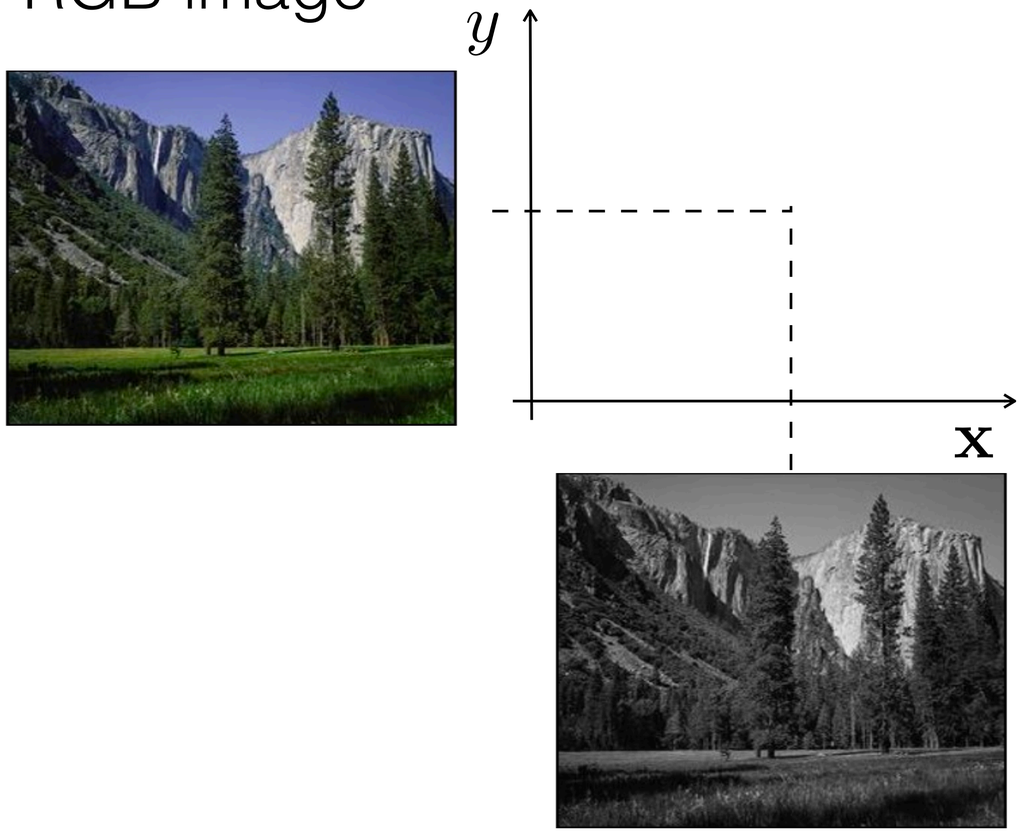




outlier

What can go wrong: **inappropriate choice of loss function**

RGB image

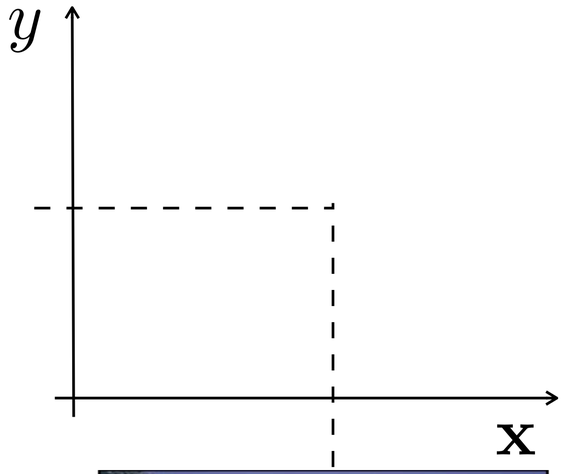


grayscale image

What can go wrong: **inappropriate choice of loss function**

Generative networks

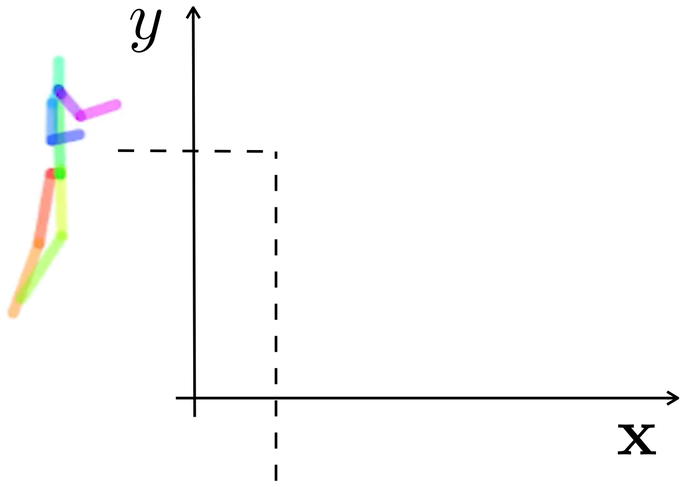
winter image



summer image

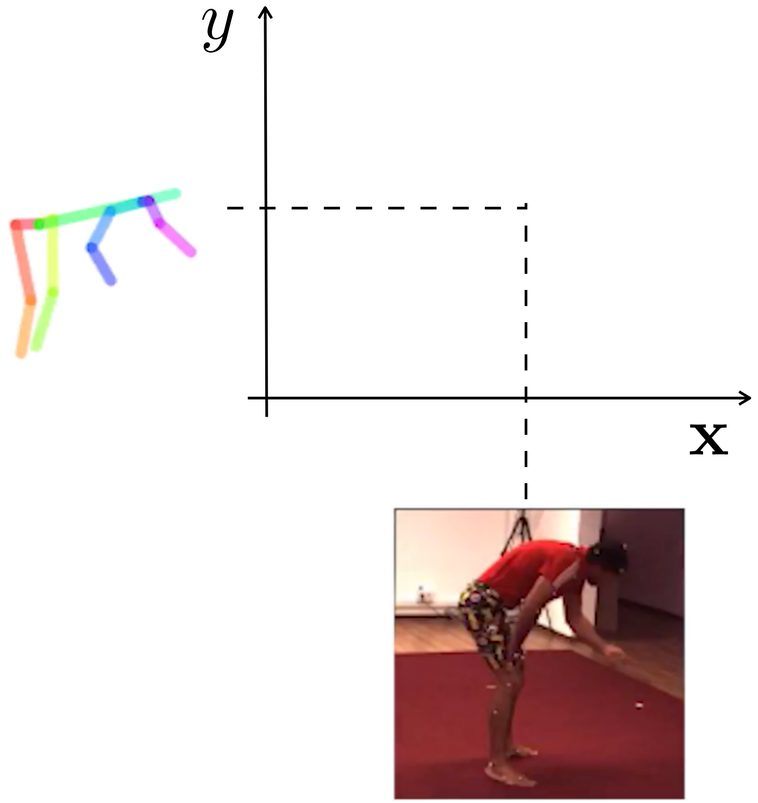
What can go wrong: **inappropriate choice of loss function**

3D pose regression



What can go wrong: **inappropriate choice of loss function**

3D pose regression



What can go wrong: **inappropriate choice of loss function**

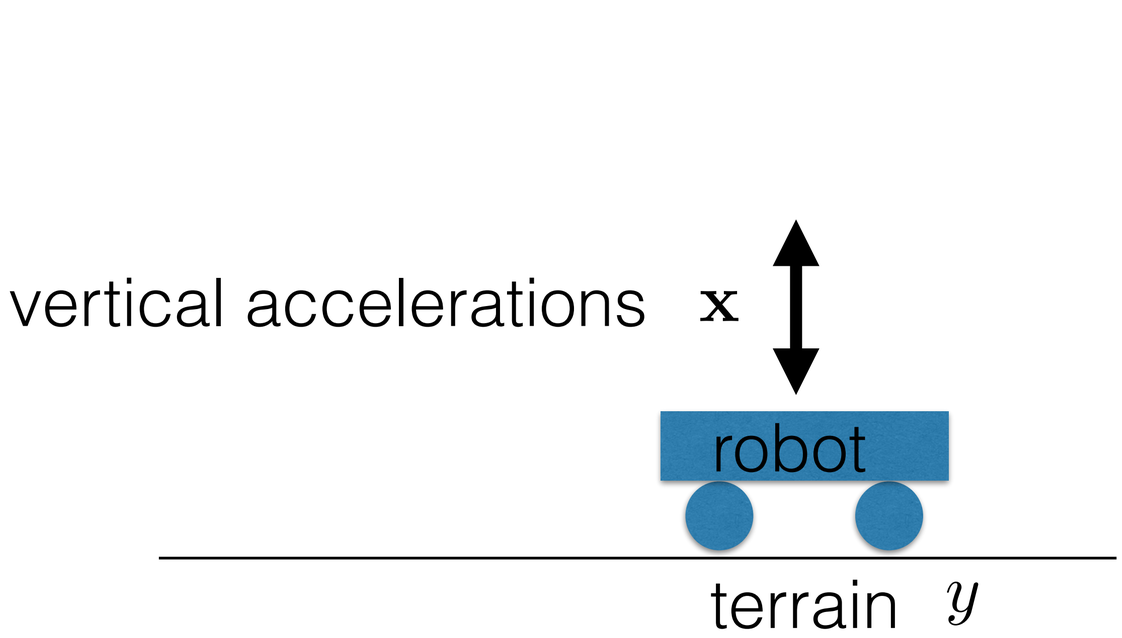
[Heess 2017] https://arxiv.org/abs/1707.02286

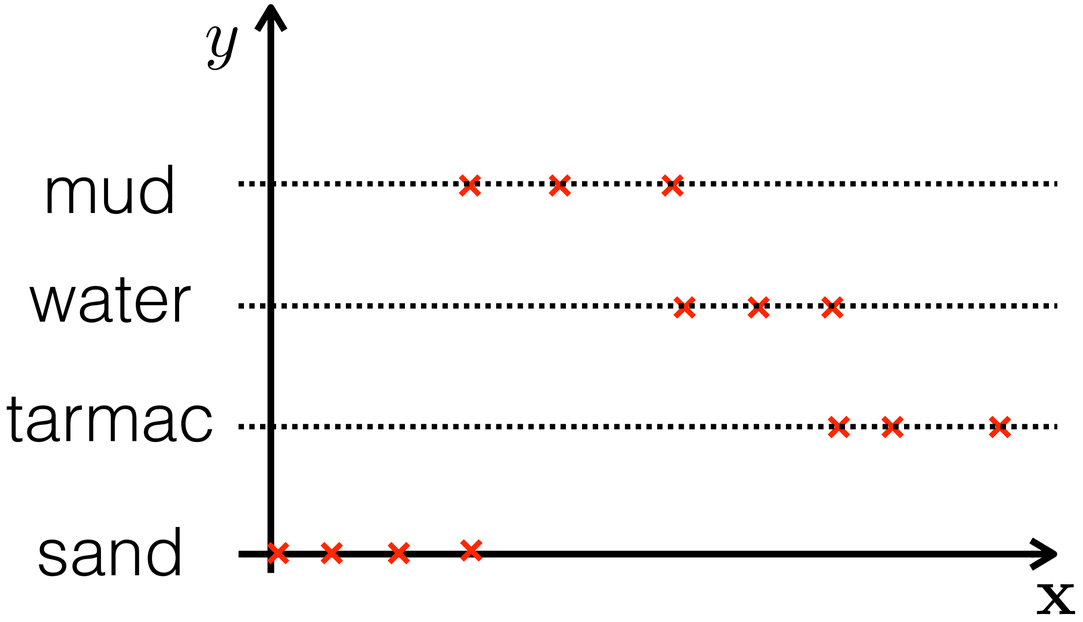
Regression

- How to create a model?
- Algorithm that maps x on y (or prob distr of y)
- This algorithm has some parameters => how to find them? => loss + trn data
- How to decide that the algorithm works well? => tst data
- What if the algorithm does not work well? What could go wrong?
  - inputs x does not allow to predict y
  - trn/tst data distribution mismatch
  - inappropriate choice of loss function (3rd lecture - MLE & loss)
  - model architecture does not generalize well (5th+ lecture - convnets)
  - learning fails to find good parameters (6th lecture - training & convergence)

Motivation example: classification

- How to create regression model?
- Is it what we really wants?

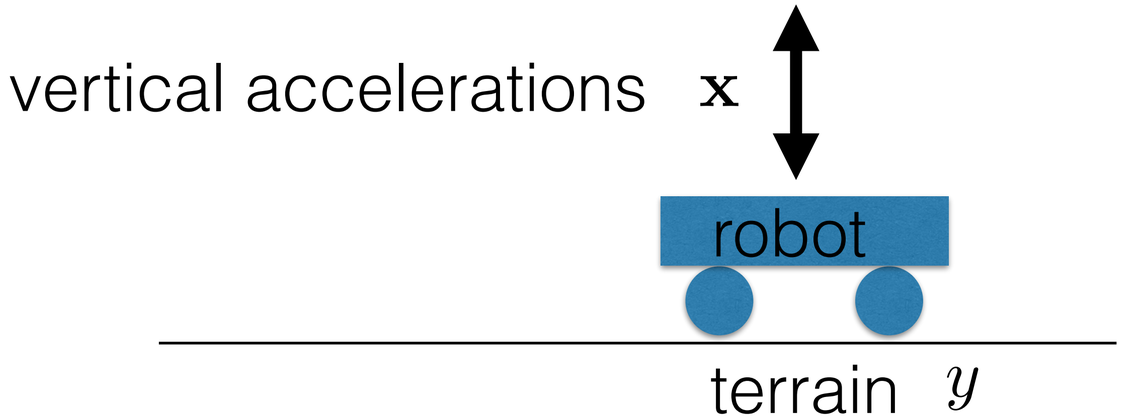


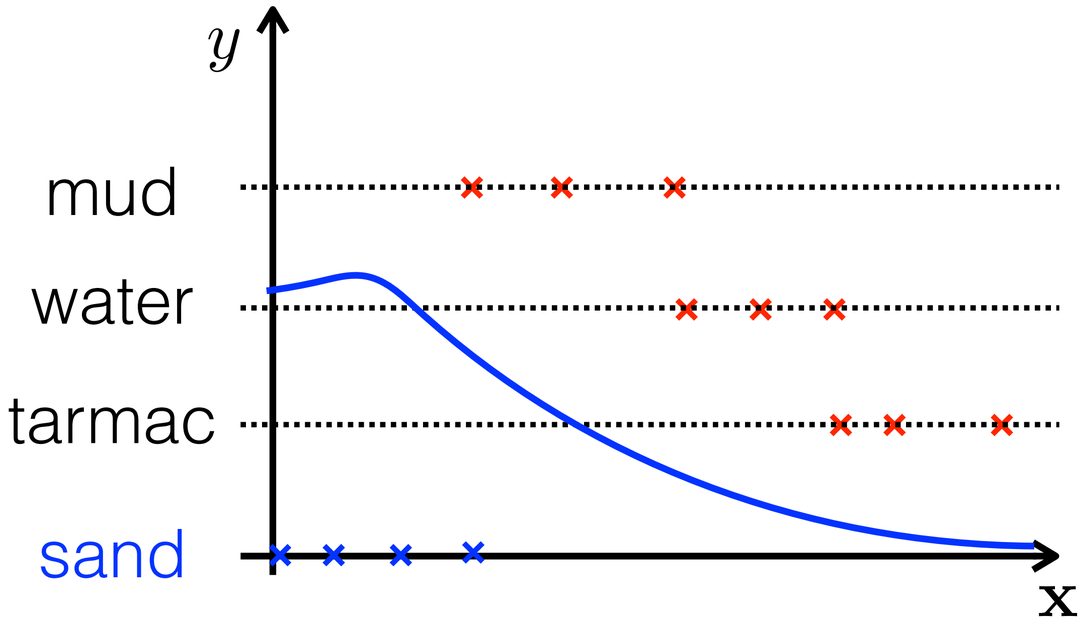


trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

Motivation example: classification

- How to create regression model?
- Is it what we really wants?

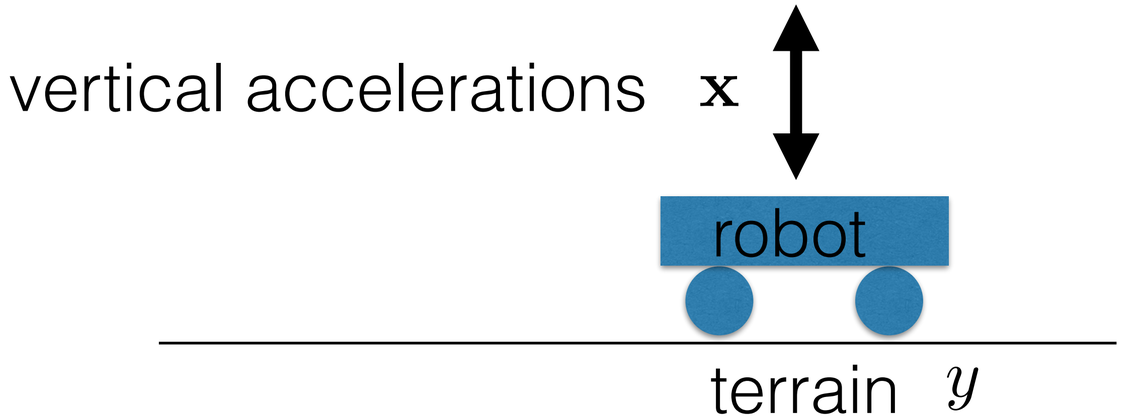


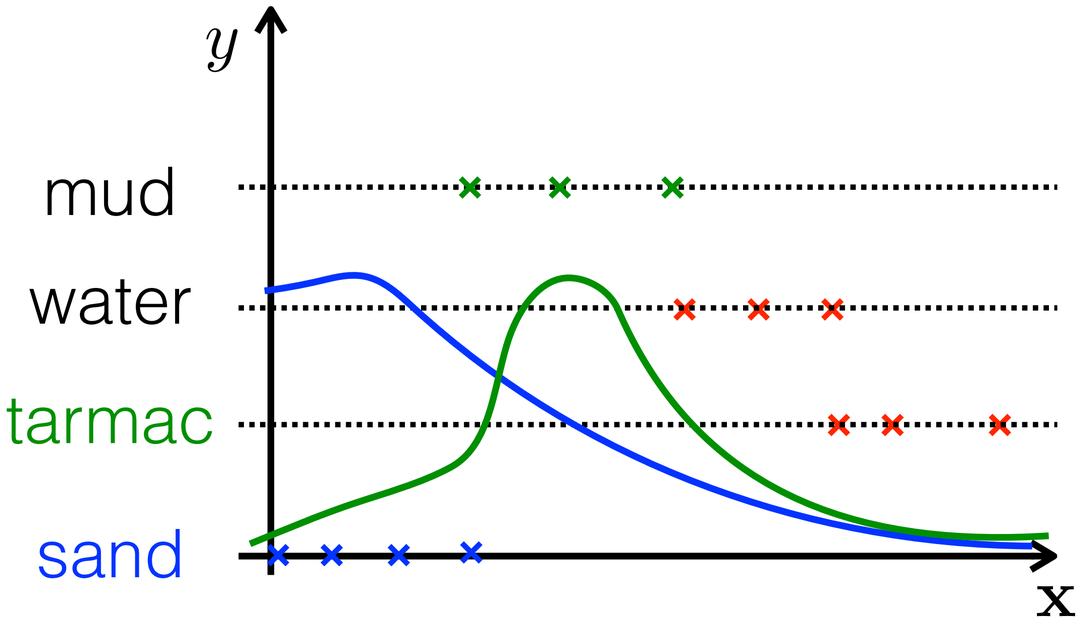


trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

Motivation example: classification

- How to create regression model?
- Is it what we really wants?

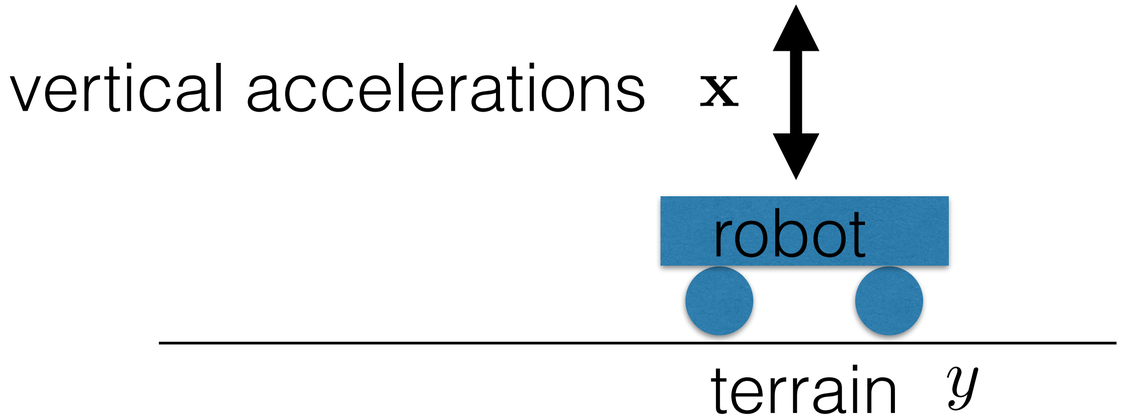


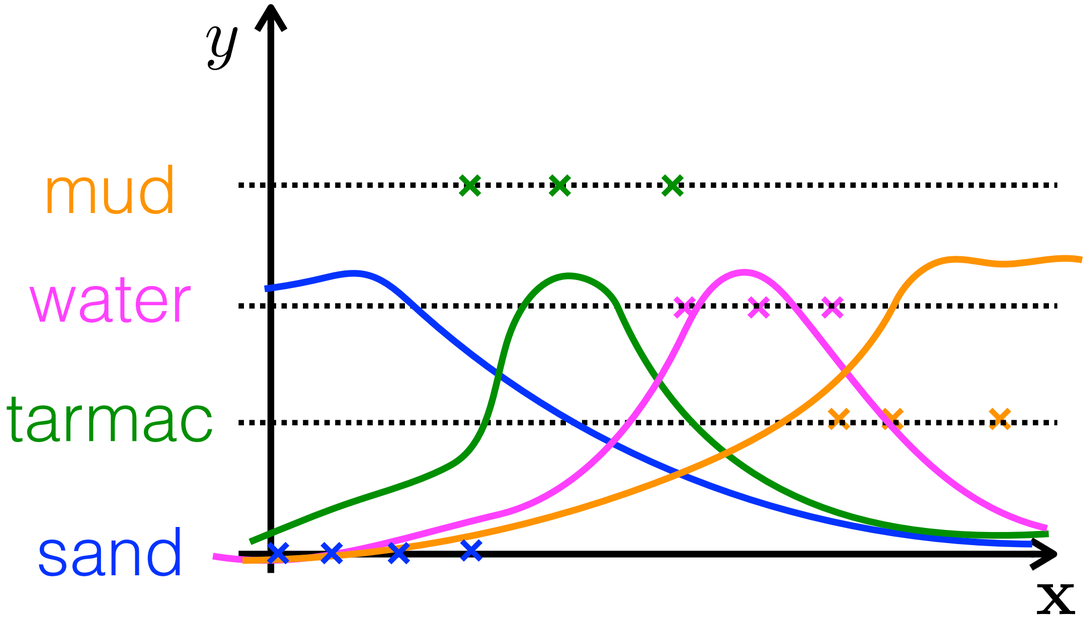


trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

Motivation example: classification

- How to create regression model?
- Is it what we really wants?

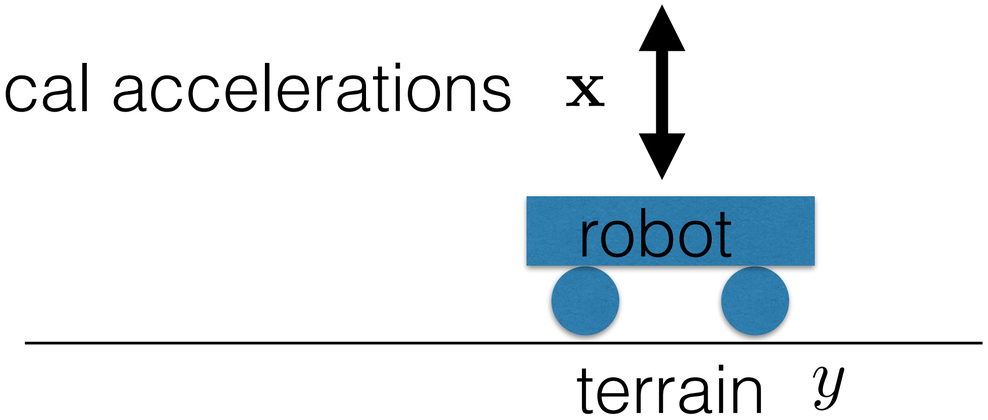


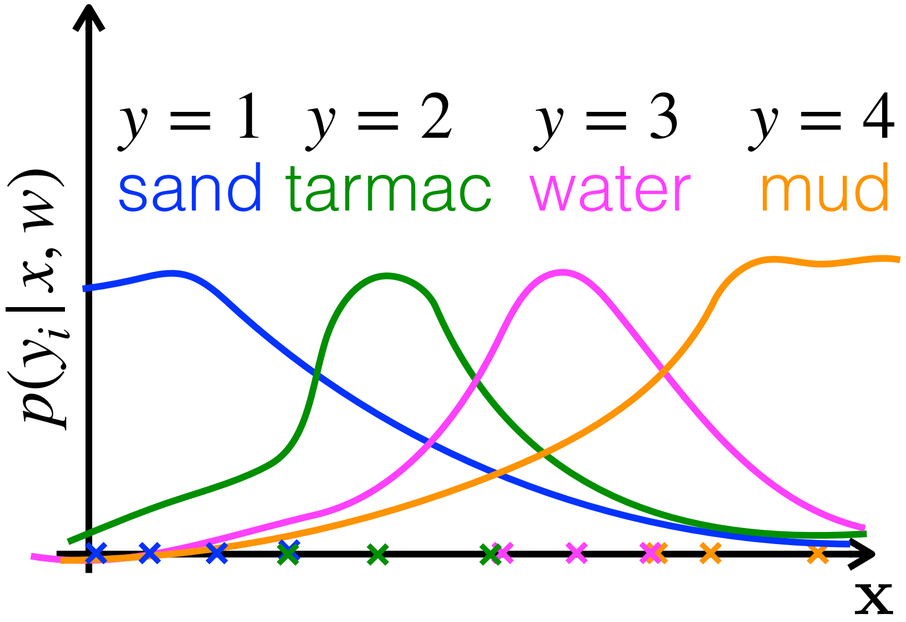


trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

Motivation example: classification

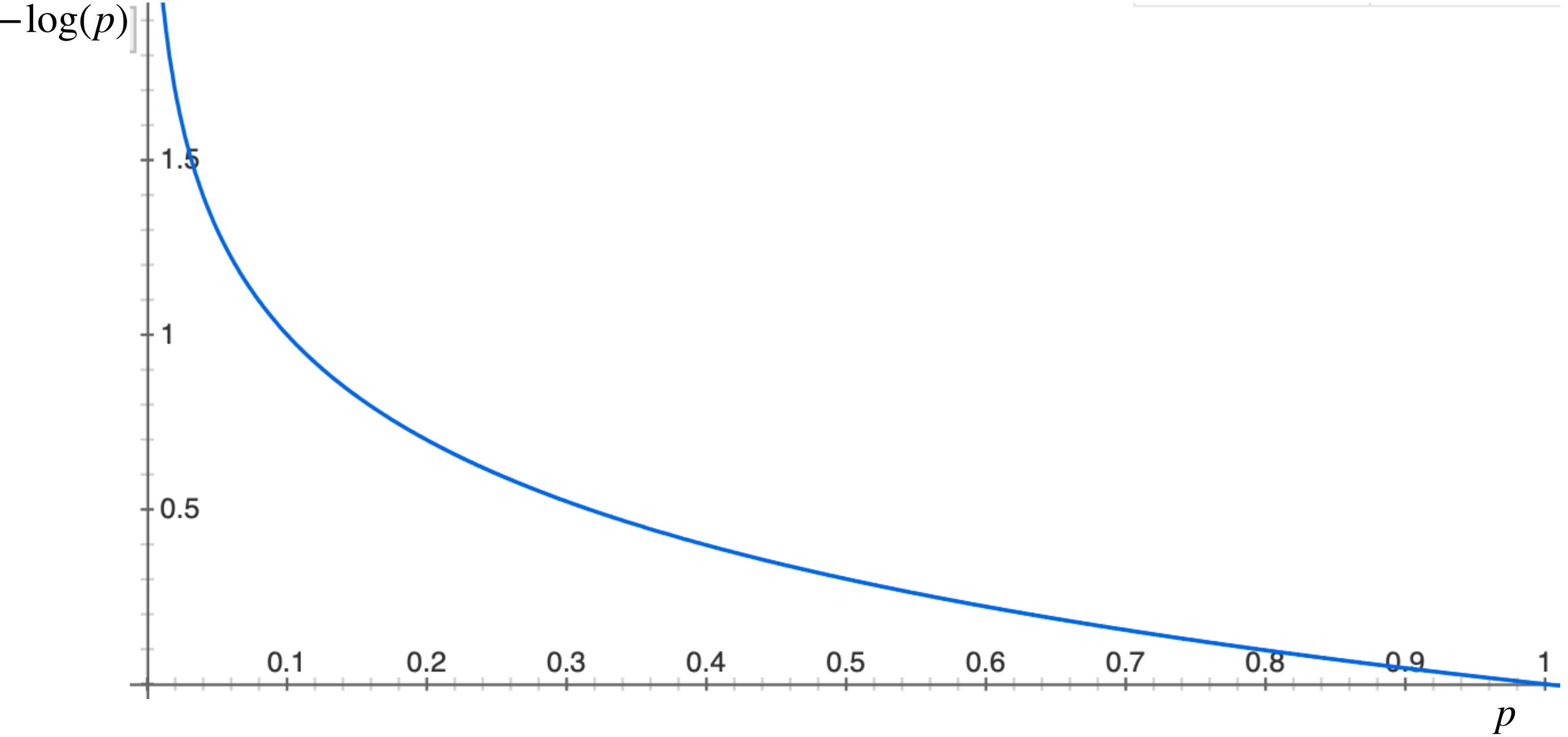
- How to create regression model?
- Is it what we really wants?
- 4 functions predicting class probabilities $p(y_1|x,w_1),\ p(y_2|x,w_2),\ \ldots$
- that sum up to one over classes for given x $\sum_i p(y_i|x,w) = 1$
- Loss pull blue function up for blue points, etc…





trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

Motivation example: classification



Competencies required for the test T1

- Model with parameters => learning
- Learning = loss + trn data + optimization procedure
- Evaluation = measuring performance (not necessary loss) on tst data
- What could go wrong?
  - inputs x does not allow to predict y
  - trn/tst data distribution mismatch
  - model does not generalize well
  - learning fails to find good parameters
  - inappropriate choice of loss function
- Regression vs Classification
- **Next lecture:** Linear classification of RGB images