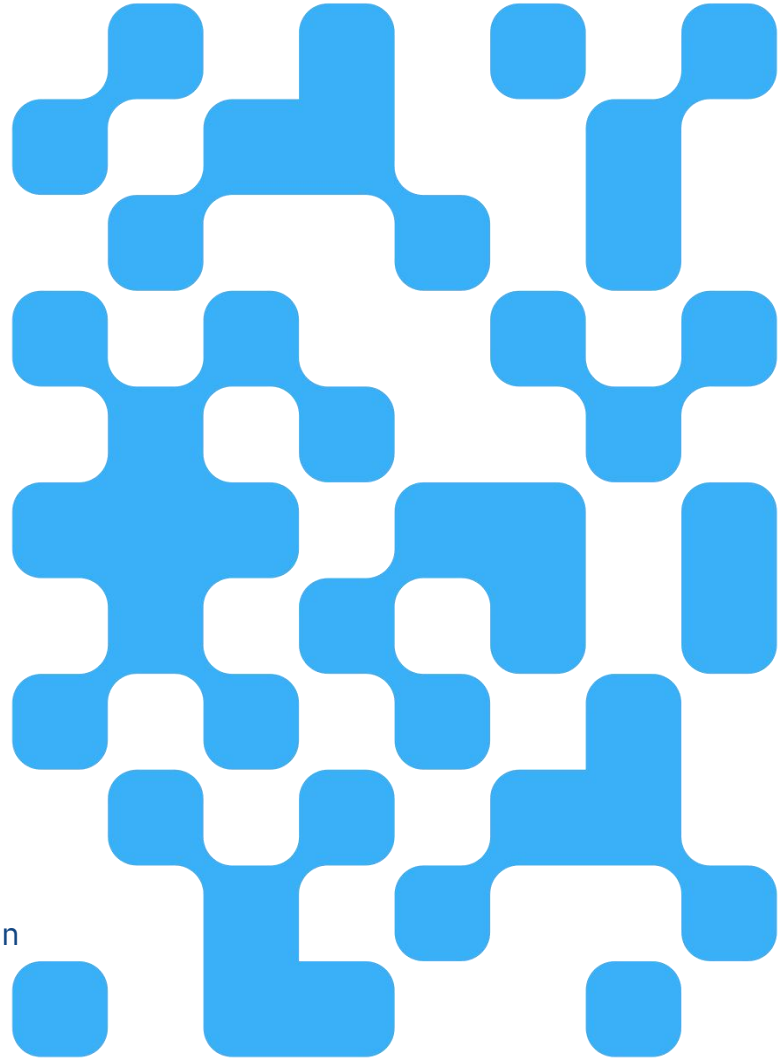


Intro to Machine Learning

2026 (ML-2026)

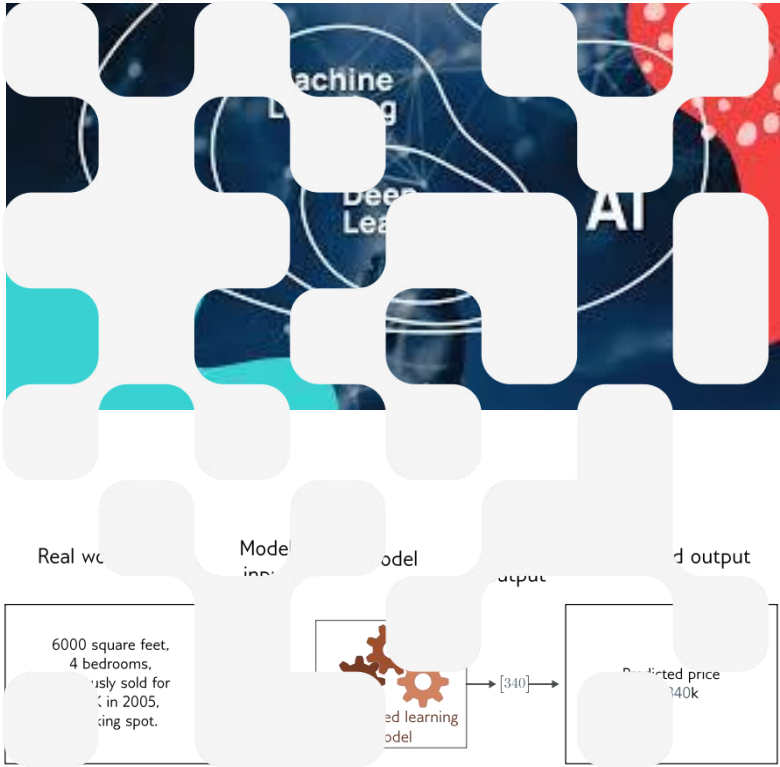
Week 01.

by Alexei Kornaev, Dr. Sc., Assoc. Prof., Innopolis University
Researcher at the RC for AI, National RC for Oncology n.a. NN Blokhin



Agenda

I. Logistics of the course
II. ML Overview: ML = E + T + P
III. Experience (E) in ML
IV. ML Tasks (T)
V. Performance measure (P) in ML
VI. Week 01 task: Linear regression
VII. MLE and MSE losses

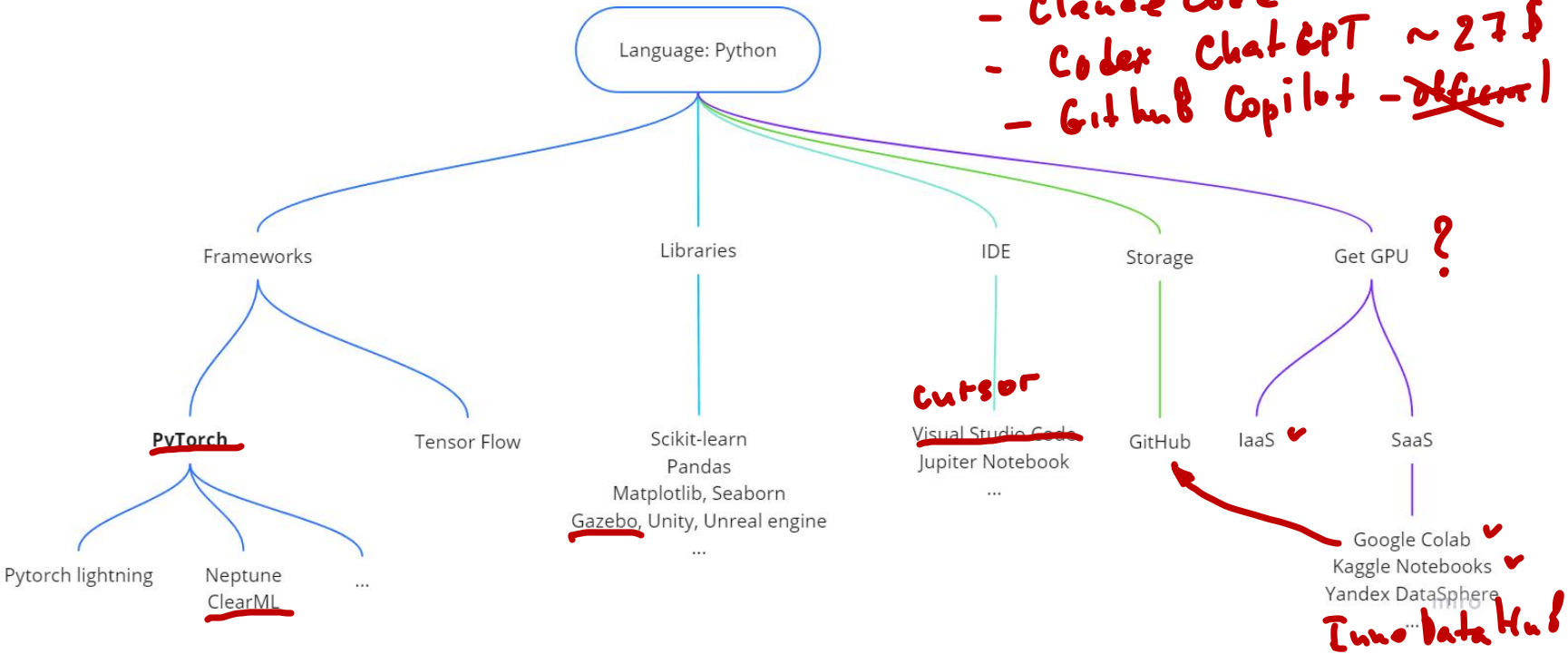


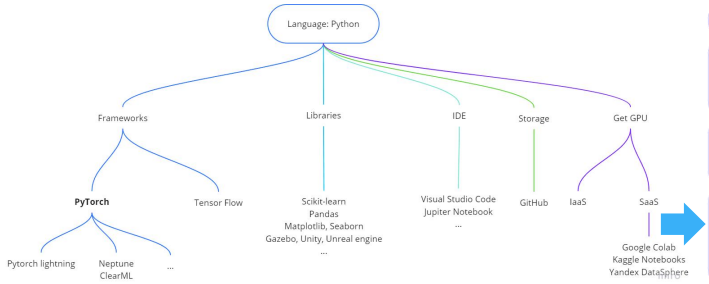
All models are wrong, but some are useful.
/George Box/

Course objectives

1. Explain how ML works: from basic ideas to real-world problems
2. Teach you how to build a model from scratch or use an open-source model to solve a problem
3. Help you take a few steps forward from educational problems to scientific ones

Syllabus


Language: Python
Frameworks
Libraries
IDE
Storage
Get GPU
PyTorch
Tensor Flow
Scikit-learn
Pandas
Matplotlib, Seaborn
Gazebo, Unity, Unreal engine
...
Visual Studio Code
Jupiter Notebook
...
GitHub
IaaS
SaaS
Pytorch lightning
Neptune
ClearML
...
Google Colab
Kaggle Notebooks
Yandex DataSphere
...
Code Agents
- Claude Code
- Codex ChatGPT ~ 27 $
- GitHub Copilot -
cursor
Inno Data Hub



Критерий	Kaggle Notebooks	AWS SageMaker	Azure ML Studio	Google Colab	Yandex DataSphere
Вычислительные ресурсы	Ограничены	Масштабируемы, без серьезных ограничений	Масштабируемы, без серьезных ограничений	Ограничены	Масштабируемы, без ограничений времени использования
Ориентация	Соревнования по машинному обучению	Промышленная разработка, машинное обучение	Промышленная разработка, машинное обучение	Обучение, исследования	Промышленная разработка, машинное обучение
Интеграция с облачными хранилищами	Автоматическое монтирование датасетов	Создание датасетов из хранилищ AWS	Создание датасетов из хранилищ Azure	Упрощенный доступ к данным через Google Drive	Создание датасетов из хранилищ Yandex Cloud Object Storage, хранение датасетов в проекте
Поддержка коллективной работы	Обмен ноутбуками и датасетами с сообществом	Общие Docker-образы, унификация сред	Общие Docker-образы, унификация сред	Поддержка совместной работы через Google Drive	Коллективная работа над проектами и ресурсами
Интерфейс пользователя	Jupyter Notebook	Платформенно-ориентированный UI	Платформенно-ориентированный UI	Jupyter Notebook	Jupyter Notebook
Интеграция с Git	Доступна	Доступна	Доступна	Доступна через интерфейс командной строки	Доступна
Подход к данным и вычислительным ресурсам	Сеансы с ограниченным временем использования	Отделение данных от вычислительных ресурсов	Отделение данных от выч	Сеансы с ограниченным временем использования, необходимость сохранения данных в Google Drive	Отделение данных от вычислительных ресурсов, возможность легкого переключения между конфигурациями VM
Особенности для образования и исследований	Соревнования по машинному обучению, обмен решениями	—	—	Широко используется в академических и образовательных целях	Оптимизация пользовательского опыта для студентов, интеграция с учебными курсами

Handbook on ML by Yandex, 2022.

Books

- Handbook on Machine Learning by M. Artemyev et al., Yandex, 2022 (in Russian)
- Understanding Deep Learning by Simon J.D. Prince, 2026
  Practical Deep Learning / FastAI book by Jeremy Howard
  Deep Learning by Ian Goodfellow and Yoshua Bengio and Aaron Courville, 2016.

Online platforms, courses, resources

Sirius online courses on ML (in Russian)
Stepik online courses (in Russian)
Hugging Face online courses
Coursera is unavailable so far

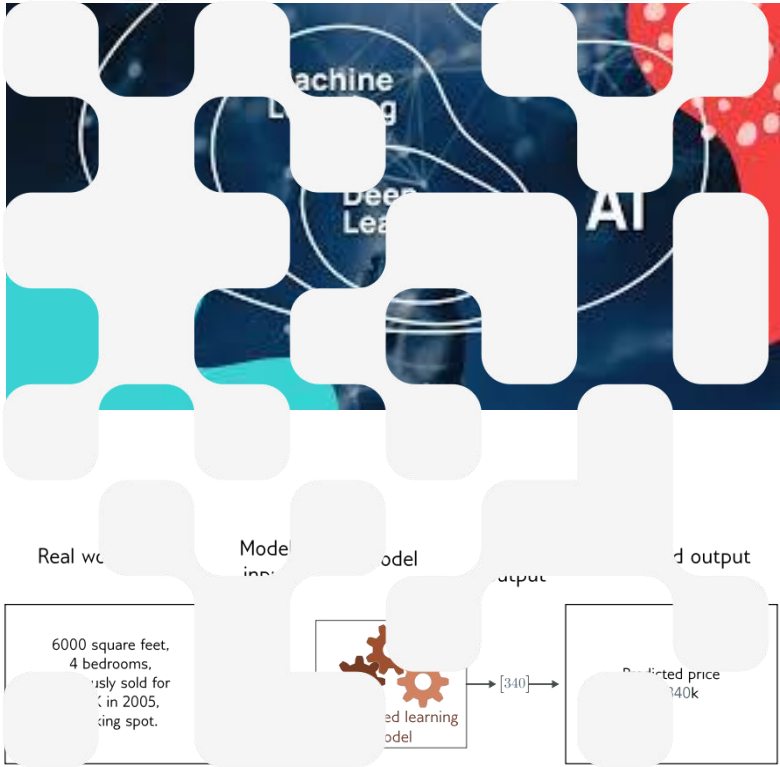
3Blue1Brown, Animated Math
PyTorch Tutorial by Patrick Loeber, 2020

#someLinks

Read here: https://arxiv.org/, https://scholar.google.ru/
Draw here: https://miro.com/app/dashboard/
Write the code here: https://colab.research.google.com/
Collect the code here: https://github.com/
Find the journal here: https://journalfinder.elsevier.com/
Find the conference here: https://portal.core.edu.au/conf-ranks/?search=A

Agenda

I. Logistics of the course
II. ML Overview: ML = E + T + P
III. Experience (E) in ML
IV. ML Tasks (T)
V. Performance measure (P) in ML
VI. Week 01 task: Linear regression
VII. MLE and MSE losses



All models are wrong, but some are useful.
/George Box/

Deterministic vs Stochastic approaches to modeling



$$\frac{d^2x}{dt^2} = -g;$$

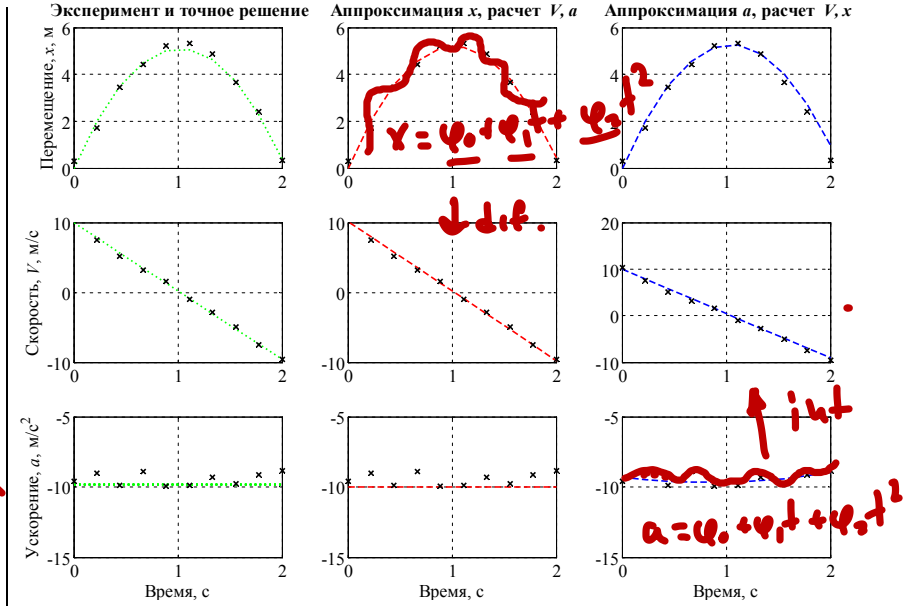
$$V(0) = V_0, \quad x(0) = x_0.$$

$$a = -g,$$

$$V = -gt + V_0,$$

$$x = -\frac{gt^2}{2} + V_0 t + x_0.$$

Deterministic approach: solution of diff. eq. + initial conditions



Stochastic approach: experiment + approximation

“When I use a word,” Humpty Dumpty said in rather a scornful tone,
“it means just what I choose it to mean—neither more nor less.”
Through the looking glass

Terms

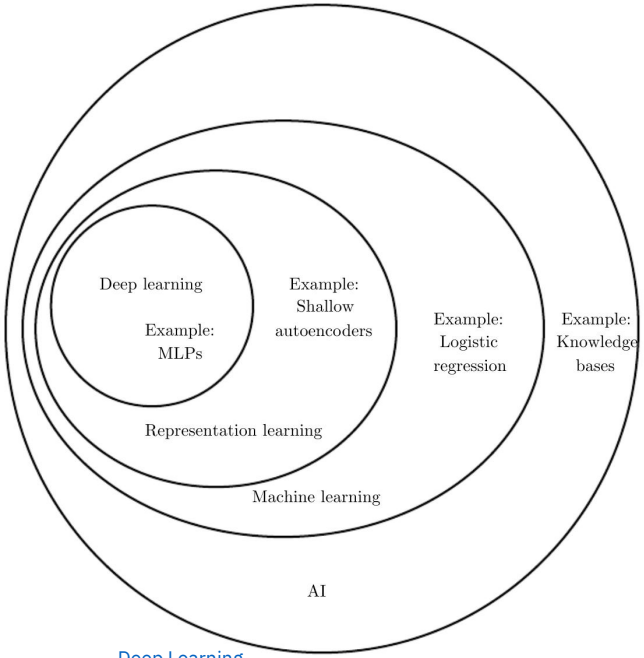
Искусственный интеллект / Artificial Intelligence (AI):

область информатики, занимающаяся моделированием разумного поведения в компьютерах / a branch of computer science dealing with the simulation of intelligent behavior in computers (Merriam-Webster)

Машинное обучение / Machine Learning (ML):

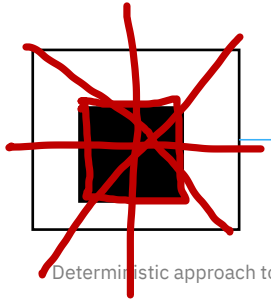
область знаний, в которой компьютеры обучаются без явного программирования/ field of study that gives computers the ability to learn without being explicitly programmed (Arthur Samuel, 1959);

задача «З», в ходе решения которой программа обучается из опыта «О» и повышает меру качества «К» / well-posed learning problem: a computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E (Tom Mitchell, 1998)



Deep Learning

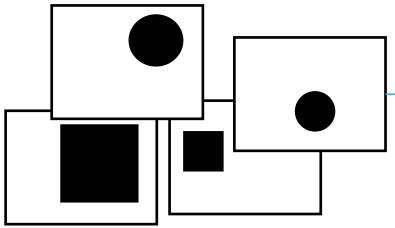
Intuition



A l g o r i t h m
Edge Detection;
Contour Construction;
Center Finding;
Axis of Symmetry Finding;

Logical Inference: This is a **square**

Deterministic approach to pattern recognition



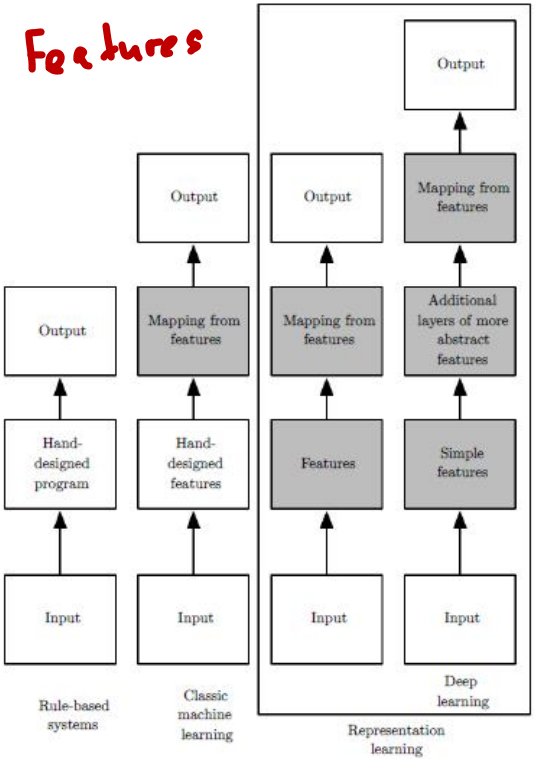
A l g o r i t h m
learn from experience E (from dataset of images), improve the performance measure P

Trained Inference: This is a **square** (not a circle)

Stochastic ML approach to pattern recognition

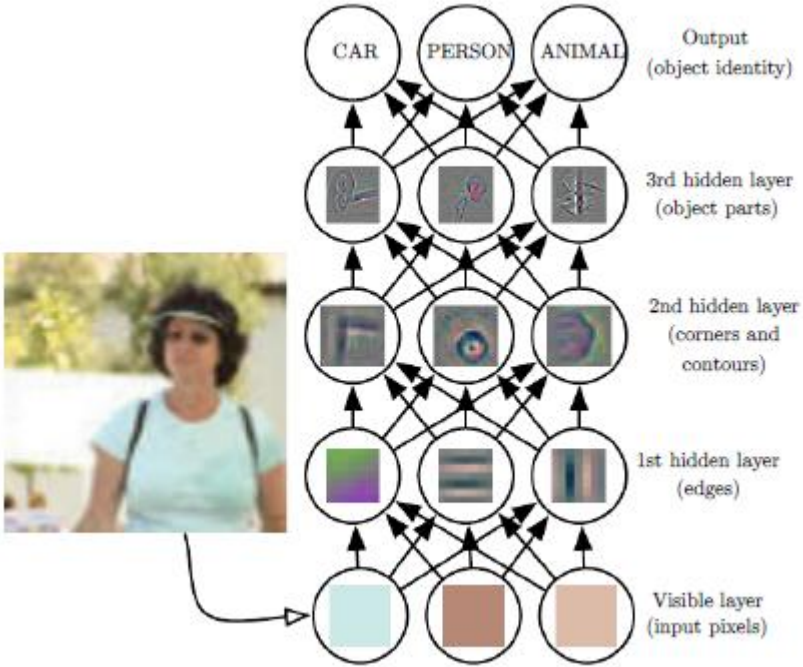
Intuition

Features



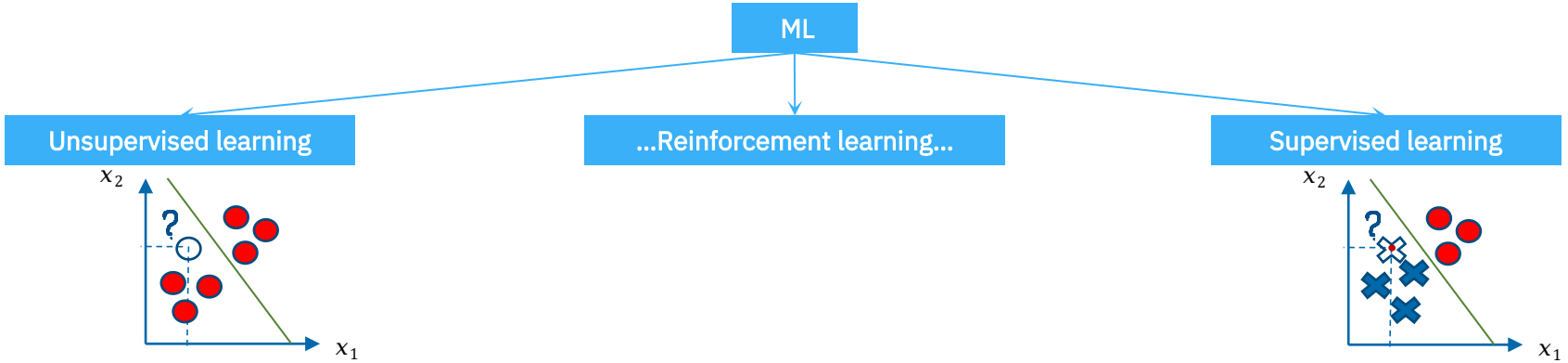
Deep Learning 2016

CNN



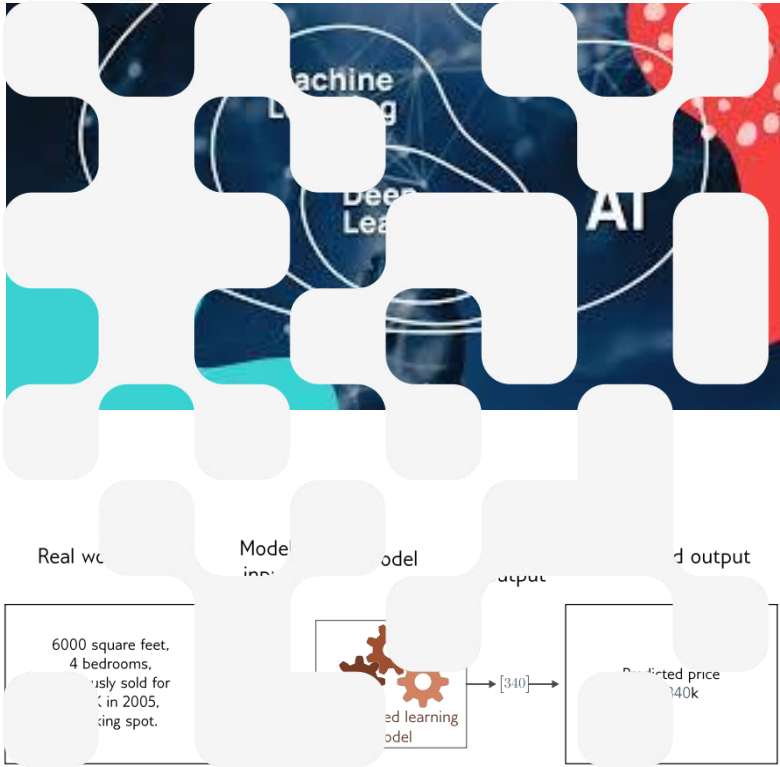
Deep Learning

Approaches to ML



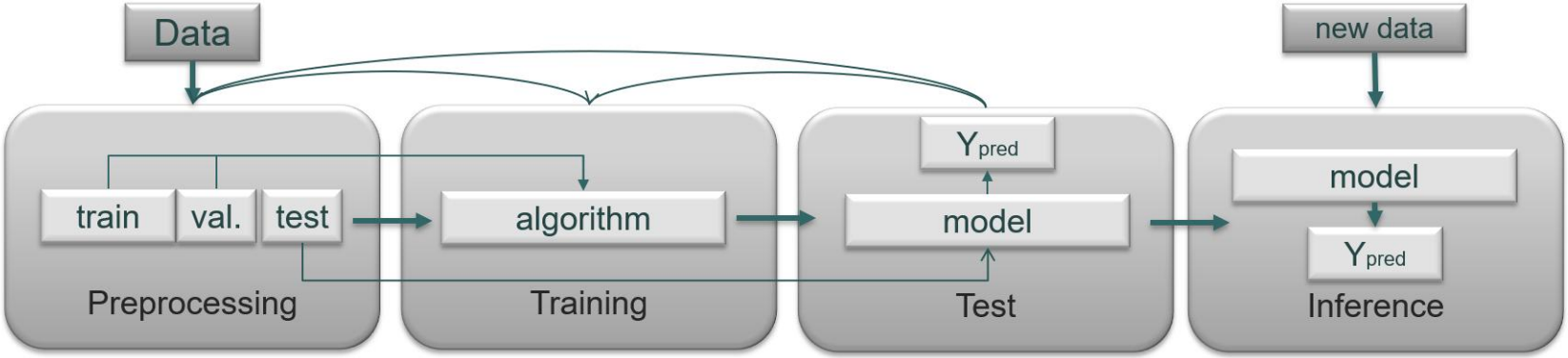
Agenda

I. Logistics of the course
II. ML Overview: ML = E + T + P
III. Experience (E) in ML
IV. ML Tasks (T)
V. Performance measure (P) in ML
VI. Week 01 task: Linear regression
VII. MLE and MSE losses



All models are wrong, but some are useful.
/George Box/

Flowchart for an ML model design (tabular data)

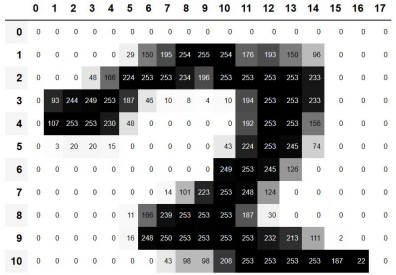


Data: input(s) is (are) stationary

king - man + woman = queen

Numeric	Visual	Textual
Feature(s)	Image	Word

Feature	Value	Descript.
x_0 …	2	# rooms
x_m	5.5	Distance



"cat"

embedding R^n

cosine similarity

$$X = x_i = \begin{bmatrix} x_0 \\ \dots \\ x_m \end{bmatrix} \qquad X = x_{ij} = \begin{bmatrix} x_{01} & \dots & x_{0n} \\ \dots & \dots & \dots \\ x_{0m} & \dots & x_{mn} \end{bmatrix} \qquad X = x_i = \begin{bmatrix} x_0 \\ \dots \\ x_m \end{bmatrix}$$

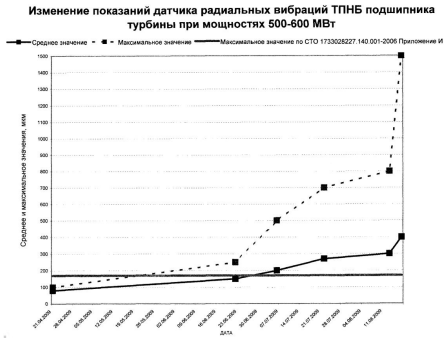
Architecture:

Perceptron, decision tree, 1D-CNN, Transformer

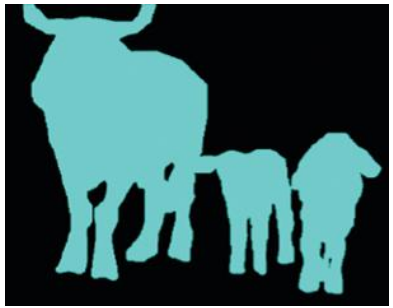
Convolutional Neural Network (CNN), Transformer

Transformer

Data: input(s) is (are) non-stationary

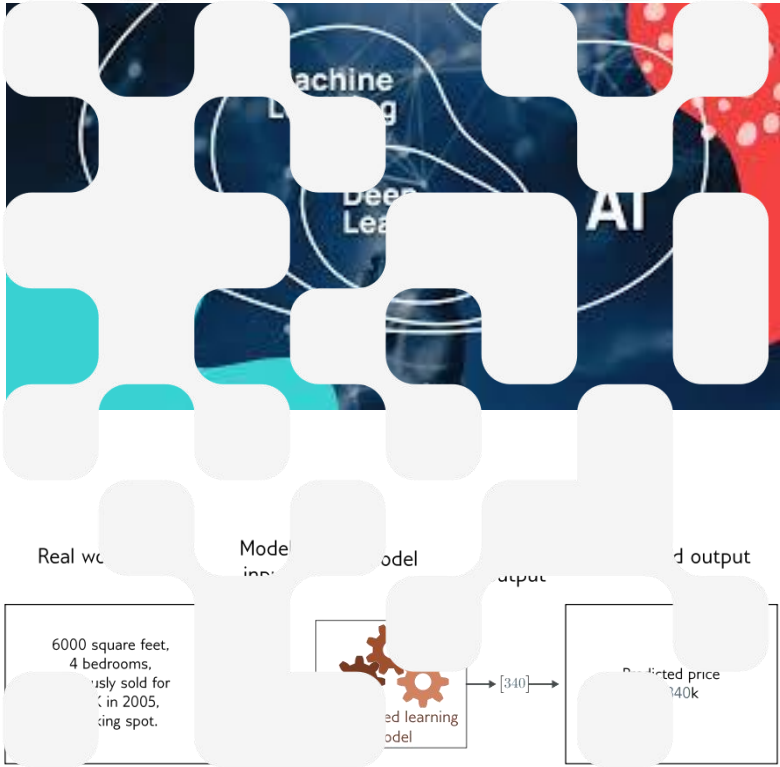
	Numeric	Visual	Textual
	Sequence of features	Sequence of images	Sequence of words “The cat's eyes sparkled with ... ”
			
Architecture:	Recurrent network (RN), Transformer	CNN + LSTM, Transformer	Transformer

Data: output(s) is (are) stationary

Numeric	Visual	Textual
Number(s)	Image	Word
Feature … **Price, MRub** x_0 … 10 … x_m …		“curiosity”
y	$Y = y_{ij} = \begin{matrix} y_{01} & \cdots & y_{0n} \\ \cdots & \cdots & \cdots \\ y_{0m} & \cdots & y_{mn} \end{matrix}$	$Y = y_i = \begin{matrix} y_0 \\ \cdots \\ y_m \end{matrix}$
Architecture:	Perceptron, decision tree, 1D-CNN, Transformer	Convolutional Neural Network (CNN) Transformer	Transformer

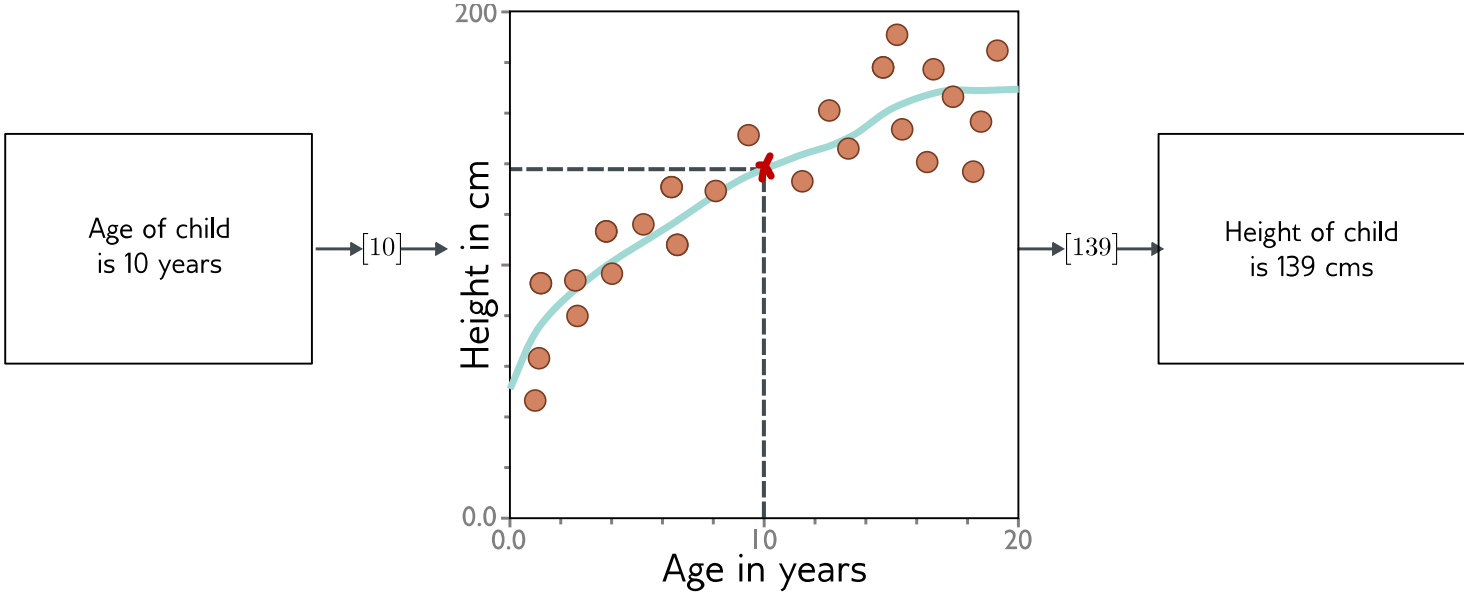
Agenda

I. Logistics of the course
II. ML Overview: ML = E + T + P
III. Experience (E) in ML
IV. ML Tasks (T)
V. Performance measure (P) in ML
VI. Week 01 task: Linear regression
VII. MLE and MSE losses



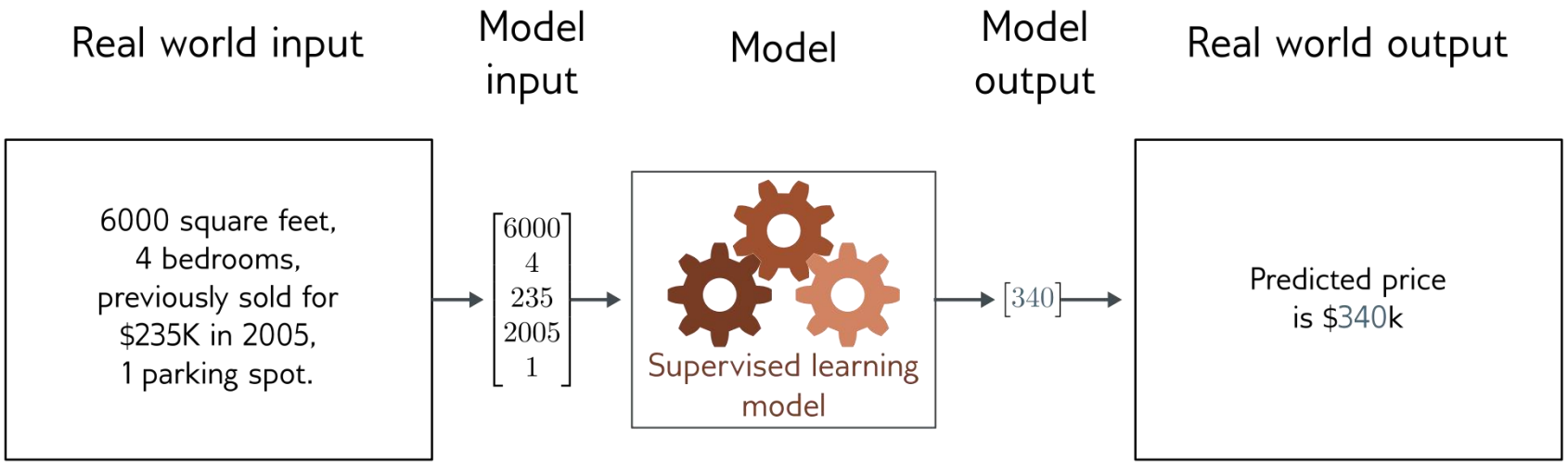
All models are wrong, but some are useful.
/George Box/

An ML model intuition



Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

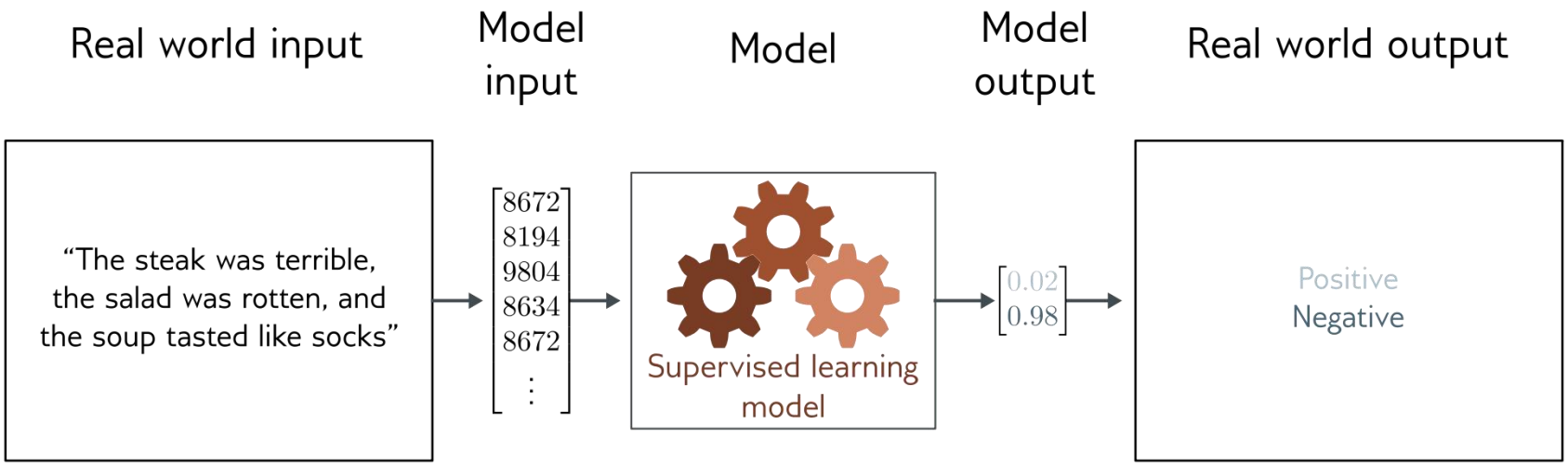
Regression



Univariate regression problem (one output, real value)
Fully connected network

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

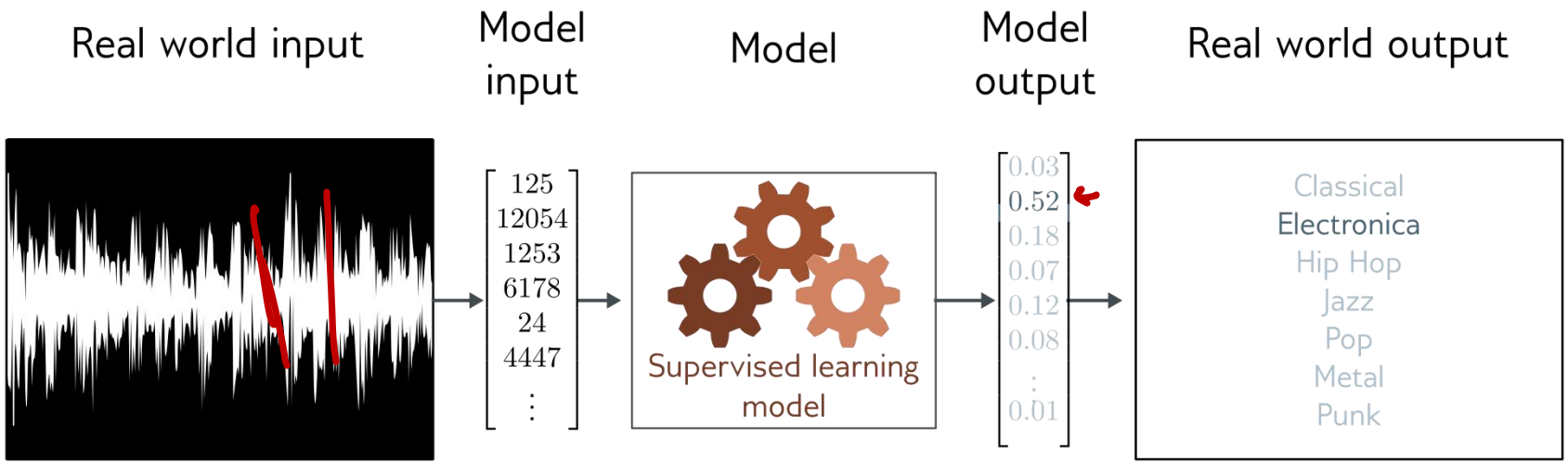
Text classification



Binary classification problem (two discrete classes)
Transformer network

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL `http://udlbook.com`.

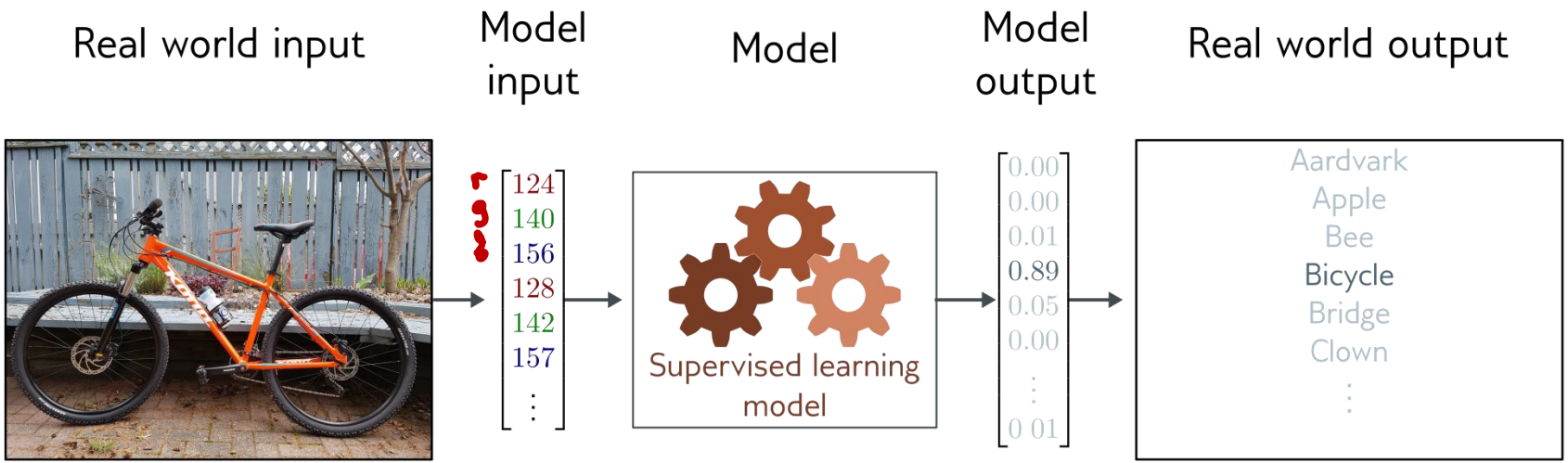
Music genre classification



Multiclass classification problem (discrete classes, >2 possible values)
Recurrent neural network (RNN), Transformers

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

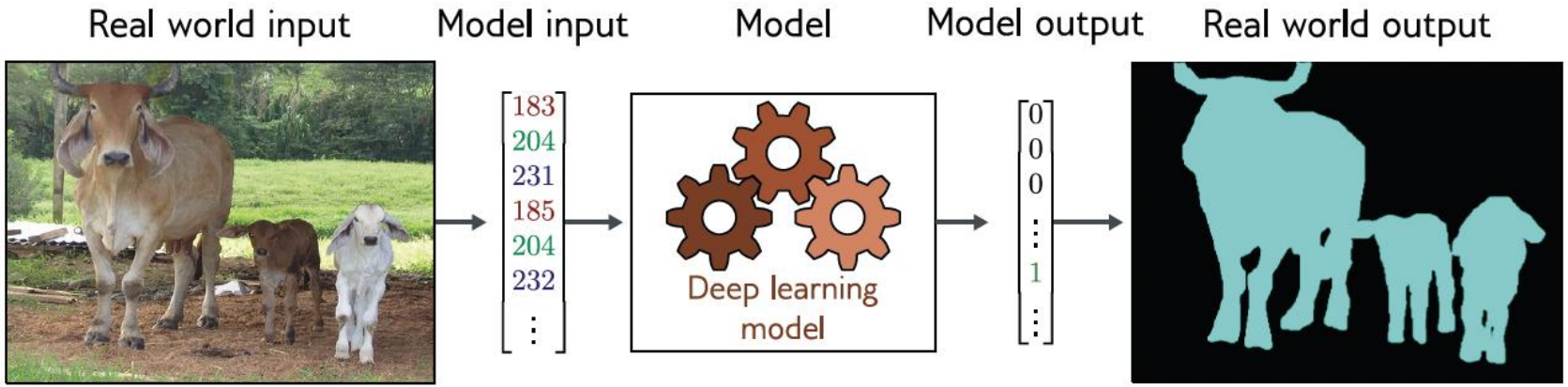
Image classification



Multiclass classification problem (discrete classes, >2 possible classes)
Convolutional networks, Transformers

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

Image segmentation

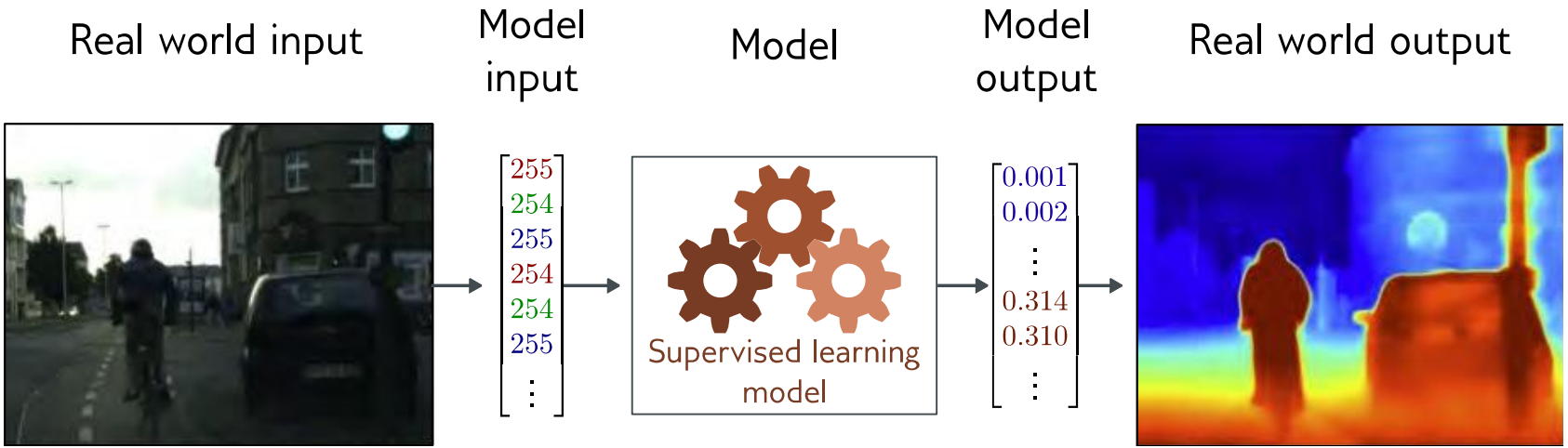


Multivariate binary classification problem (many outputs, two discrete classes)
Convolutional encoder-decoder network, Transformers

Supervised learning intuition

S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

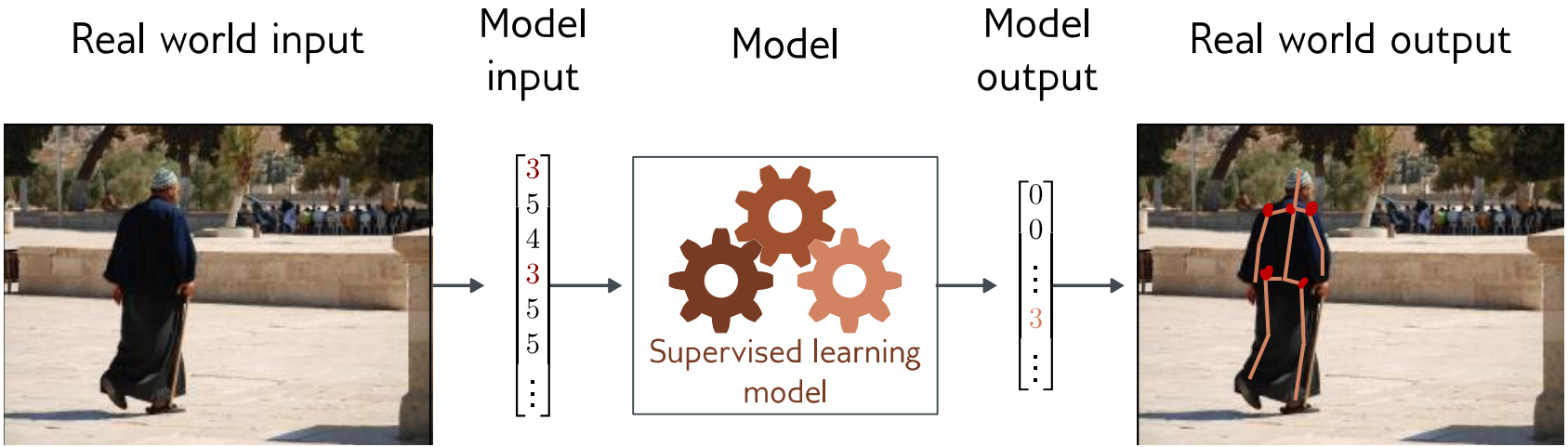
Depth estimation



Multivariate regression problem (many outputs, continuous)
Convolutional encoder-decoder network, Transformers

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

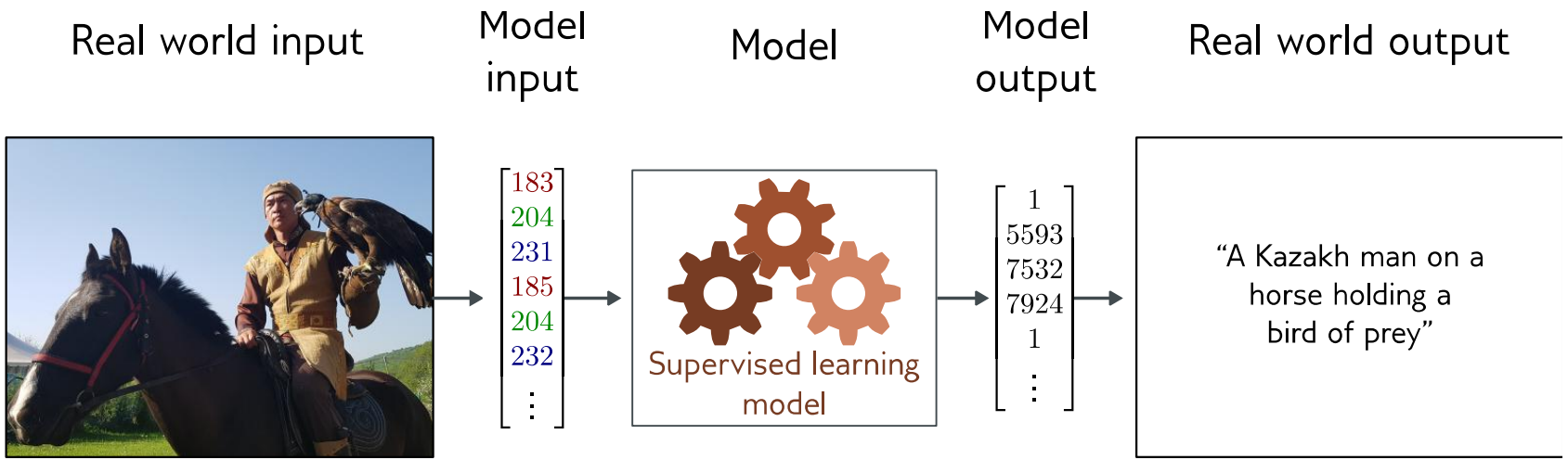
Pose estimation



Multivariate regression problem (many outputs, continuous)
Convolutional encoder-decoder network, Transformers

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

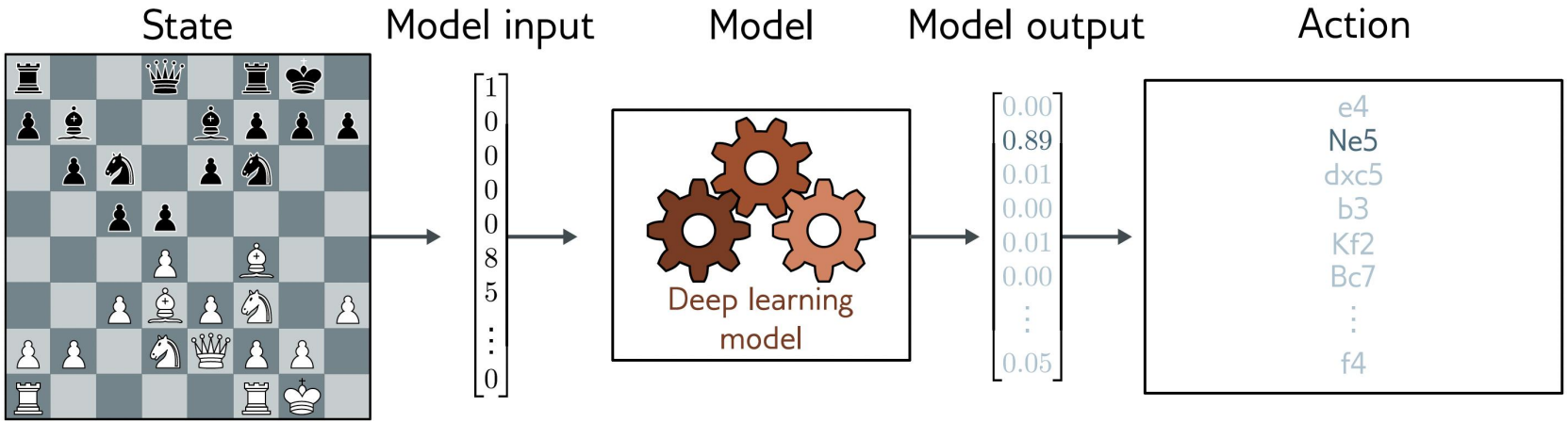
Image captioning



Visual data analysis and text generation problem
Transformer networks

Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

Games

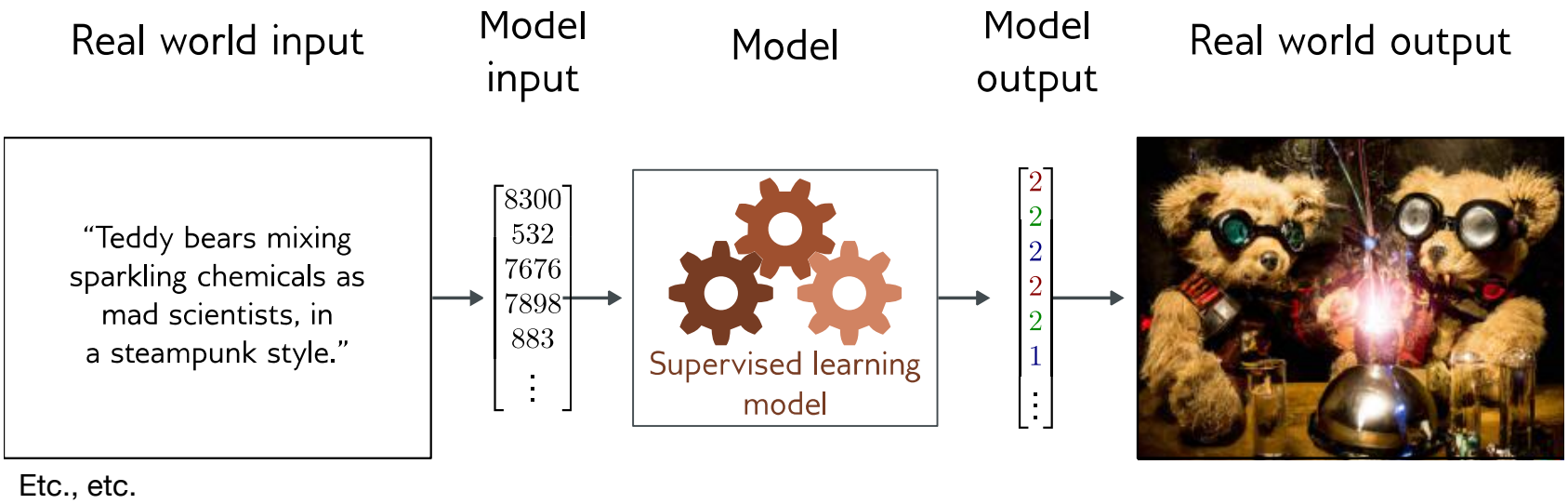


Decision making problem
Feedforward networks, CNNs, transformer networks

Supervised learning intuition

S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

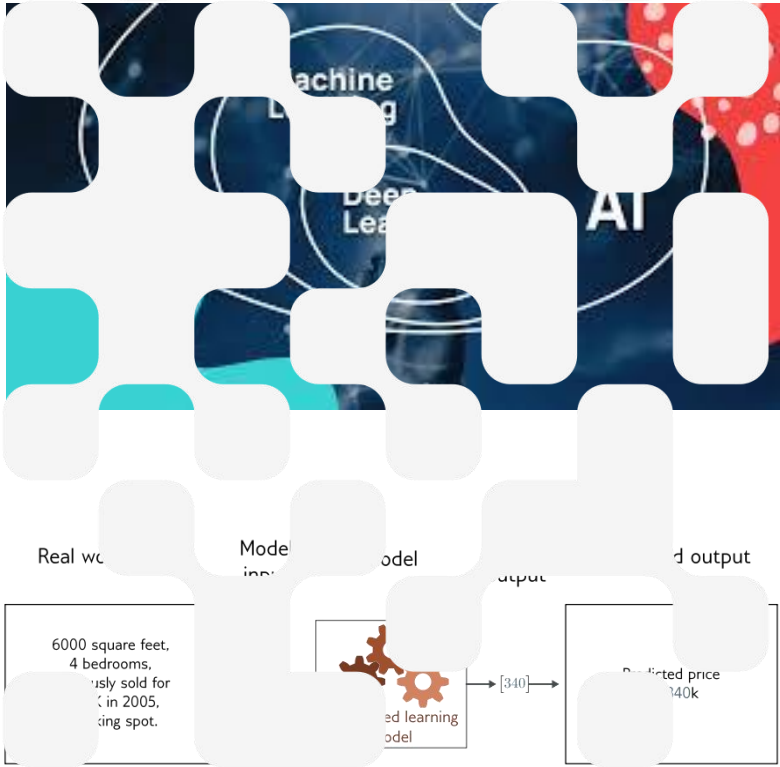
Image generation from text



Supervised learning intuition
S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

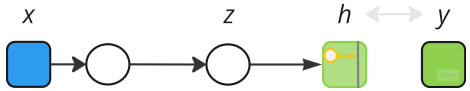
Agenda

I. Logistics of the course
II. ML Overview: ML = E + T + P
III. Experience (E) in ML
IV. ML Tasks (T)
V. Performance measure (P) in ML
VI. Week 01 task: Linear regression
VII. MLE and MSE losses



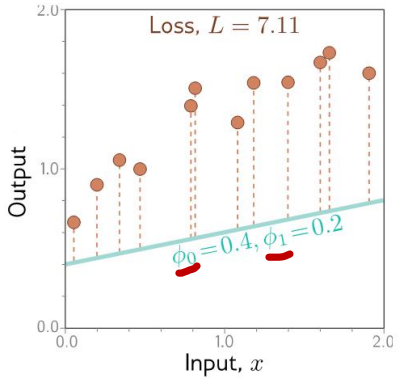
All models are wrong, but some are useful.
/George Box/

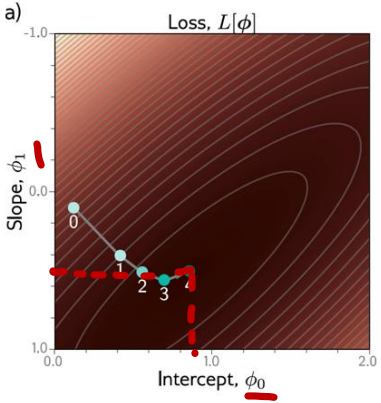
Performance Measure intuition

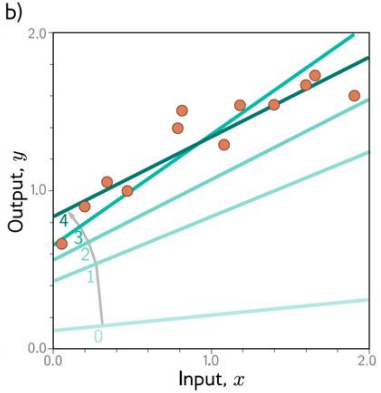


$$L_R = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - h^{(i)}\right)^2$$

Model predicts output *h* given input *x*

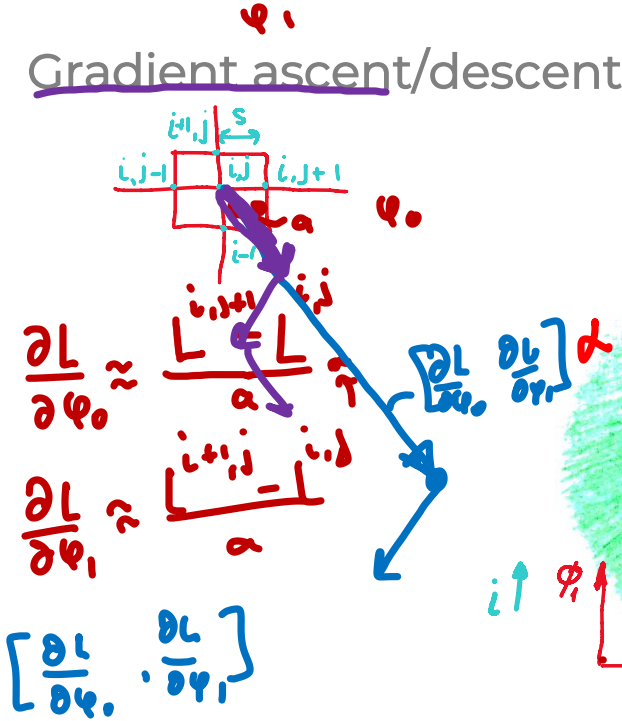


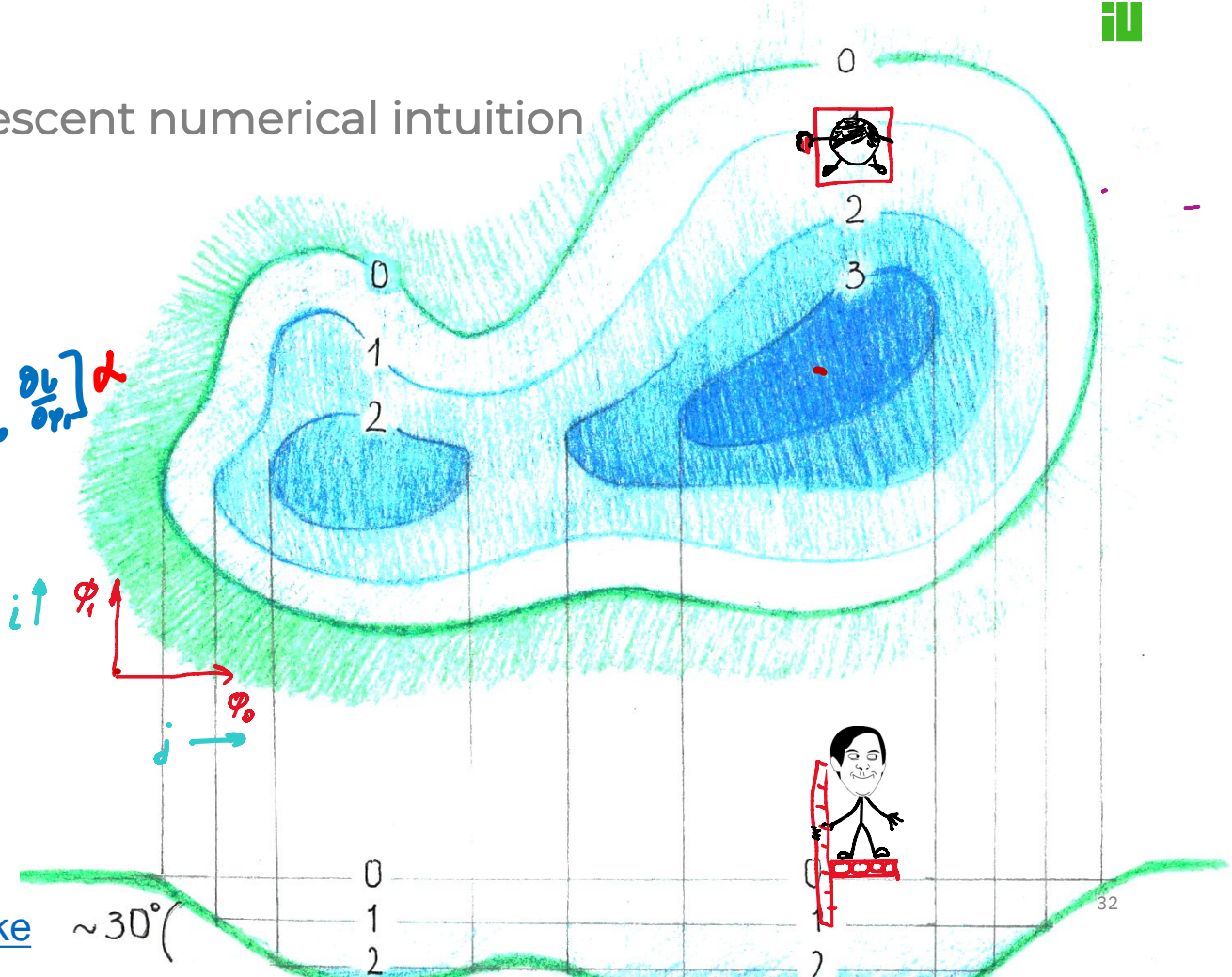




Supervised learning intuition: S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

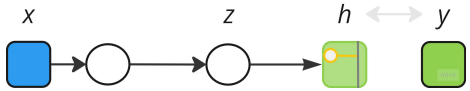
Gradient ascent/descent numerical intuition





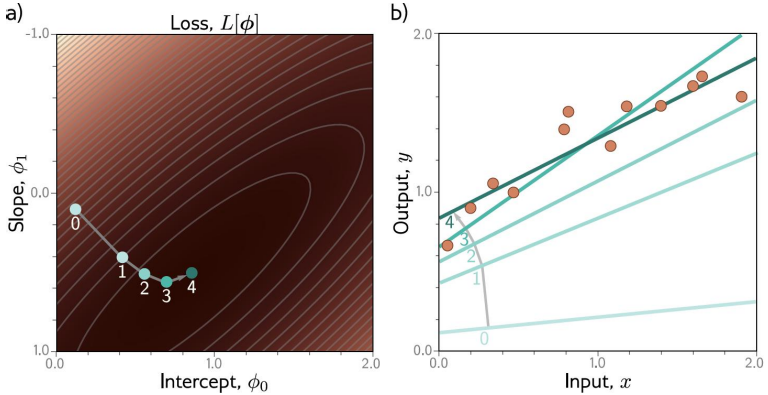
A lake $\sim 30°$

Performance Measure intuition



$$L_R = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - h^{(i)}\right)^2$$

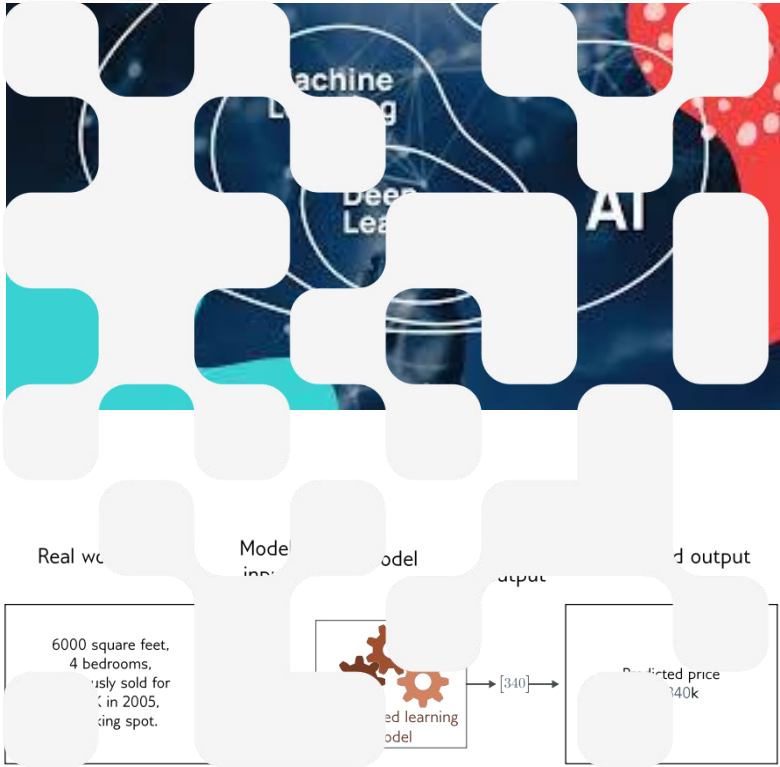
Model predicts output h given input x



Supervised learning intuition: S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

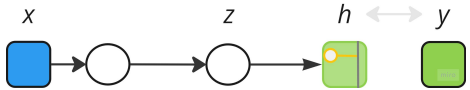
Agenda

I. Logistics of the course
II. ML Overview: ML = E + T + P
III. Experience (E) in ML
IV. ML Tasks (T)
V. Performance measure (P) in ML
VI. Week 01 task: Linear regression
VII. MLE and MSE losses



All models are wrong, but some are useful.
/George Box/

Linear Regression



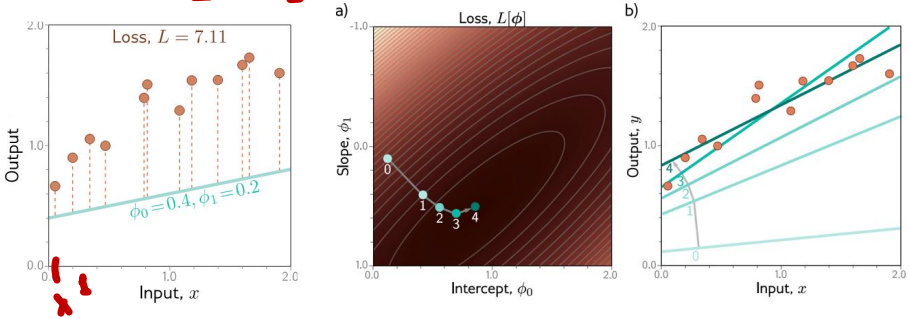
Model predicts output h given input x

Consider a model $\boldsymbol{f} = x^{(i)}$, $\boldsymbol{\phi}$ parameterized with weights $\boldsymbol{\phi}$ that maps each i-th input sample $x^{(i)}$ into the output $z^{(i)}$ which then transforms into the hypothesis $h^{(i)}$ that should be close to the label $y^{(i)}$.

$$\boldsymbol{x} = \begin{bmatrix} x^{(1)} \\ \dots \\ x^{(m)} \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ \dots \\ y^{(m)} \end{bmatrix} \boldsymbol{\phi} = \begin{bmatrix} \phi_0 \\ \phi_1 \end{bmatrix}$$

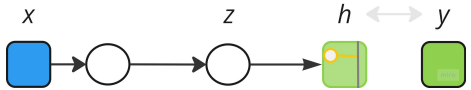
$$L(\boldsymbol{\phi}) = \frac{1}{2m} \sum_{i=1}^{m} \left(h^{(i)} - y^{(i)} \right)^2 \Rightarrow \min.$$

$h^{(i)} = \phi_0 + \phi_1 x^{(i)}$



Supervised learning intuition: S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

Linear Regression



Model predicts output *h* given input *x*

Consider a model $\boldsymbol{f} = x^{(i)}$, $\boldsymbol{\phi}$ parameterized with weights $\boldsymbol{\phi}$ that maps each i-th input sample $x^{(i)}$ into the output $z^{(i)}$ which then transforms into the hypothesis $h^{(i)}$ that should be close to the label $y^{(i)}$.

$$\boldsymbol{x} = \begin{bmatrix} x^{(1)} \\ \dots \\ x^{(m)} \end{bmatrix} \rightarrow \begin{bmatrix} 1 & x^{(1)} \\ \dots & \dots \\ 1 & x^{(m)} \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ \dots \\ y^{(m)} \end{bmatrix} \boldsymbol{\phi} = \begin{bmatrix} \phi_0 \\ \phi_1 \end{bmatrix}$$

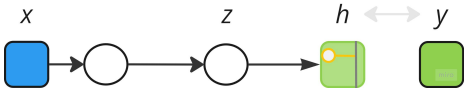
$$L(\boldsymbol{\phi}) = \frac{1}{2m} \sum_{i=1}^{m} \left(h^{(i)} - y^{(i)} \right)^2 \Rightarrow \min.$$

$$\nabla L = \frac{\partial L}{\partial \phi_j} = \frac{1}{m} \boldsymbol{x}^T (\boldsymbol{h} - \boldsymbol{y}).$$

Training algorithm.

1. Initialize the *weights* $\boldsymbol{\phi}$ with a random seed
2. Calculate the *hypothesis* matrix $\boldsymbol{h} = \boldsymbol{x}\boldsymbol{\phi}$ and the *loss gradient*: $\nabla L = \frac{1}{m} \boldsymbol{x}^T (\boldsymbol{h} - \boldsymbol{y})$
3. For the given ϕ_j components (annotated with idex 'prev', ϕ_j^{prev}) calculate the newer ones ϕ_j^{next} moving towards the direction, which is opposite to the loss gradient vector, with steps which are proportional to the *learning rate* α:
$\boldsymbol{\phi}^{next} = \boldsymbol{\phi}^{prev} - \alpha \nabla L$, or in the scalar form $\phi_0^{\text{next}} = \phi_0^{prev} - \alpha \frac{\partial L}{\partial \phi_0}$, $\phi_1^{\text{next}} = \phi_1^{prev} - \alpha \frac{\partial L}{\partial \phi_1}$
4. Repeat pp. 2-3 until the minimum of the loss function L is reached, based on the condition of small changes in its value over several neighboring iterations or based on the condition of reaching the maximum number of iterations : $L^{next} - L^{prev} < \delta$, #iter. > max # of iter
5. Save the trained model (model weights): $\boldsymbol{\phi}$.

Linear Regression. Generalization (multiple var., polynomial)



Model predicts output h given input x

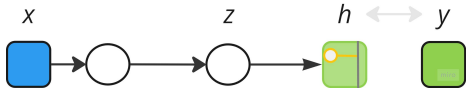
Consider a model $\boldsymbol{f} = x^{(i)}$, $\boldsymbol{\phi}$ parameterized with weights $\boldsymbol{\phi}$ that maps each i-th input sample $x^{(i)}$ into the output $z^{(i)}$ which then transforms into the hypothesis $h^{(i)}$ that should be close to the label $y^{(i)}$.

$$L(\boldsymbol{\phi}) = \frac{1}{2m}\sum_{i=1}^{m} h^{(i)} - y^{(i)}\ ^2 \Rightarrow \min.$$

Training algorithm.

1. Initialize $\boldsymbol{\phi}$
2. Calculate $\boldsymbol{h} = \boldsymbol{x}\boldsymbol{\phi}$ and $\nabla L = \frac{1}{m}\boldsymbol{x}^T(\boldsymbol{h} - \boldsymbol{y})$
3. Update $\boldsymbol{\phi}^{next} = \boldsymbol{\phi}^{prev} - \alpha\nabla L$
4. Repeat pp. 2-3 $L^{next} - L^{prev} < \delta$, #iter. > max # of iter
5. Save the trained model (model weights): $\boldsymbol{\phi}$.

Linear Regression. Generalization (multiple var., polynomial)



Model predicts output *h* given input *x*

Consider a model $\boldsymbol{f} = x^{(i)}$, $\boldsymbol{\phi}$ parameterized with weights $\boldsymbol{\phi}$ that maps each i-th input sample $x^{(i)}$ into the output $z^{(i)}$ which then transforms into the hypothesis $h^{(i)}$ that should be close to the label $y^{(i)}$.

$$L(\boldsymbol{\phi}) = \frac{1}{2m}\sum_{i=1}^{m} h^{(i)} - y^{(i)}{}^{2} \Rightarrow \min.$$

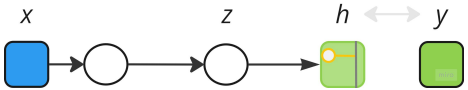
$h = \varphi_0 + \varphi_1 x + \varphi_2 x^2 + \dots$

$x_1 \quad x_2 \quad \dots$

Training algorithm.

1. Initialize $\boldsymbol{\phi}$
2. Calculate $\boldsymbol{h} = \boldsymbol{x}\boldsymbol{\phi}$ and $\nabla L = \frac{1}{m}\boldsymbol{x}^T(\boldsymbol{h} - \boldsymbol{y})$
3. Update $\boldsymbol{\phi}^{next} = \boldsymbol{\phi}^{prev} - \alpha\nabla L$
4. Repeat pp. 2-3 $L^{next} - L^{prev} < \delta$, #iter. > max # of iter
5. Save the trained model (model weights): $\boldsymbol{\phi}$.

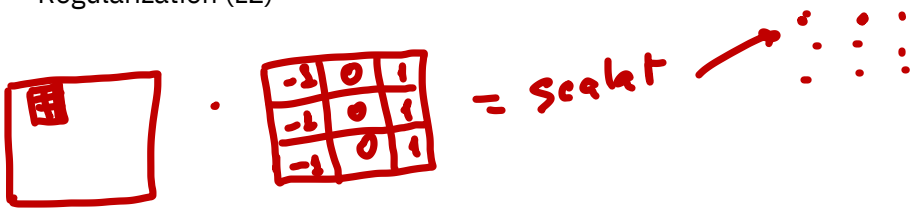
ML Settings



Model predicts output *h* given input *x*

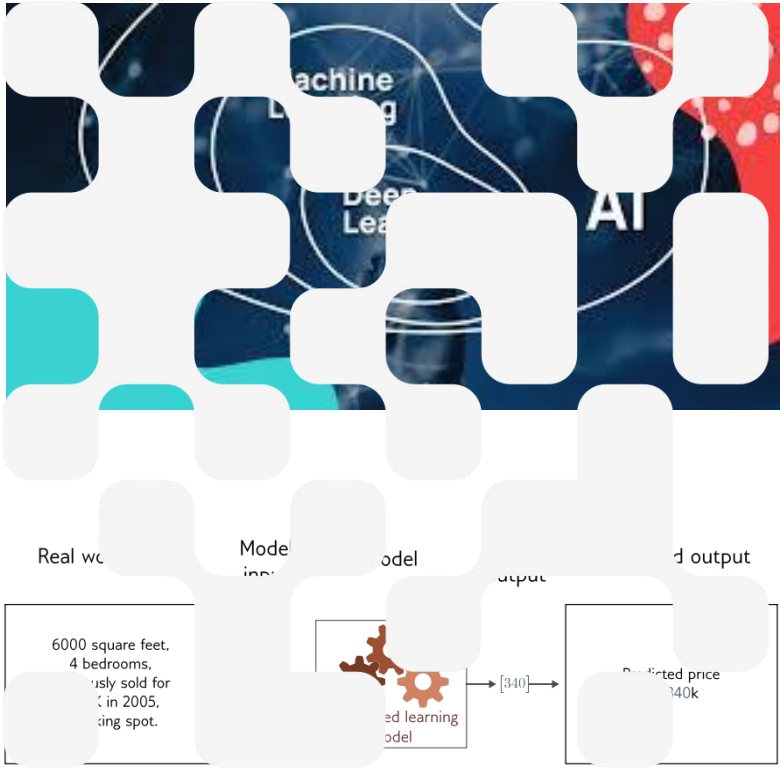
1. Feature Scaling
2. **Learning Rate**
3. **Error and # of iterations**
4. Regularization (L2)

Model *parameters* are determined during the solution of the ML problem. For example, in regression problems, the parameters are the components of the matrix of weights $\boldsymbol{\phi}$. *Hyperparameters* are set by the user, usually not in a single way, and their values affect the values of the sought parameters.

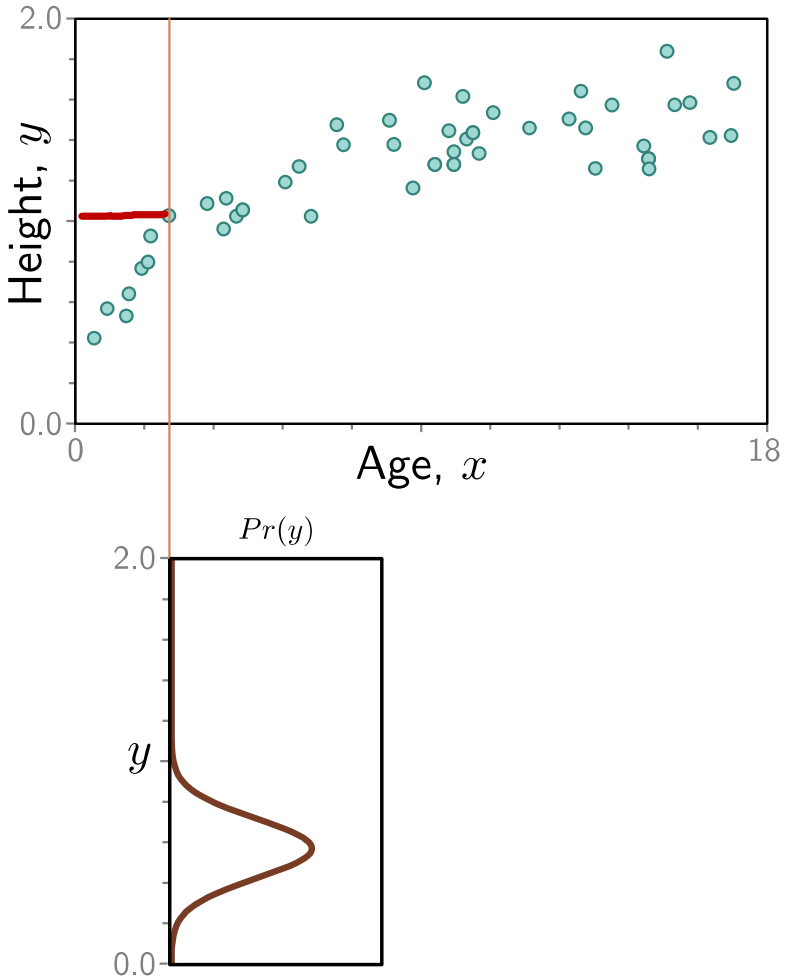


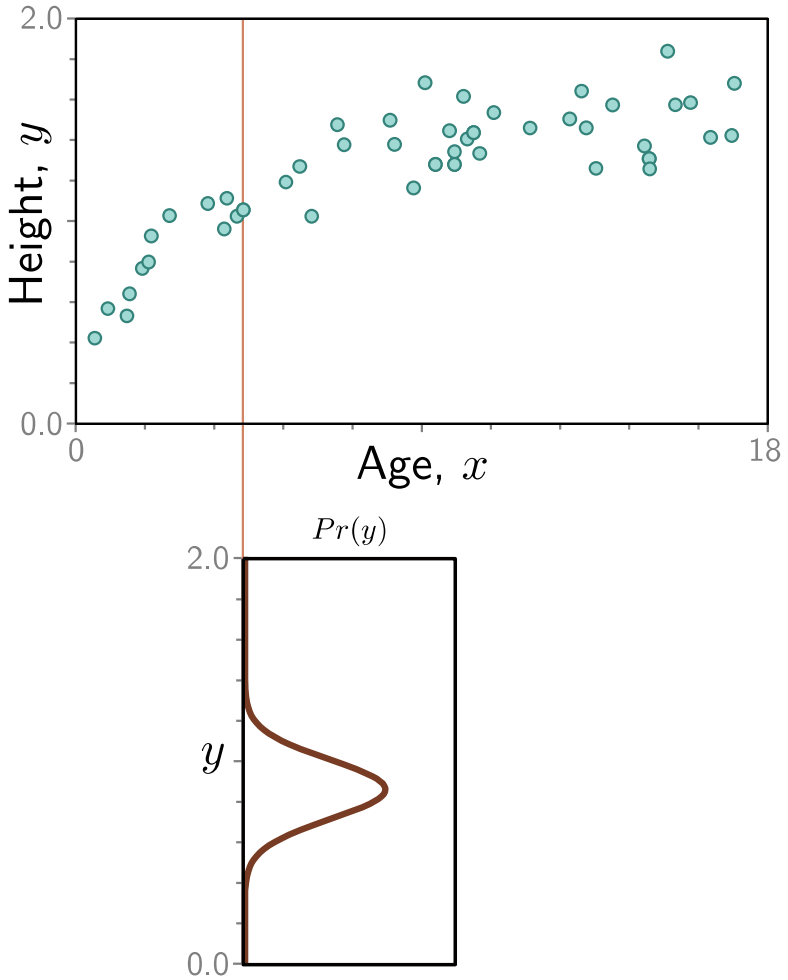
Agenda

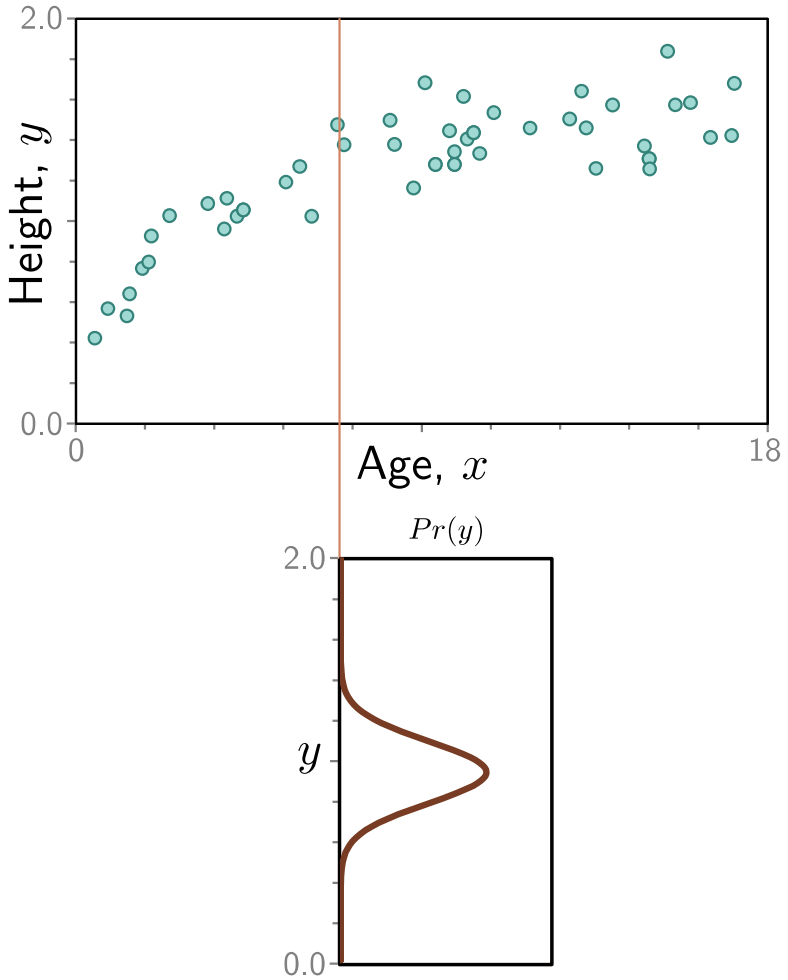
I. Logistics of the course
II. ML Overview: ML = E + T + P
III. Experience (E) in ML
IV. ML Tasks (T)
V. Performance measure (P) in ML
VI. Week 01 task: Linear regression
VII. MLE and MSE losses

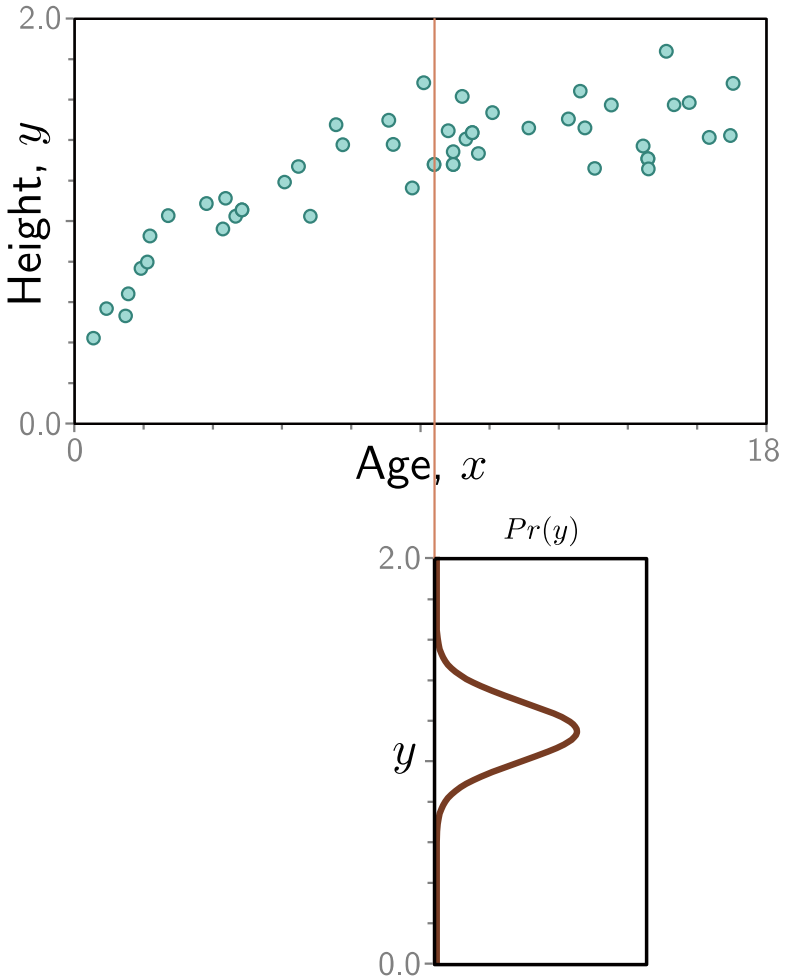


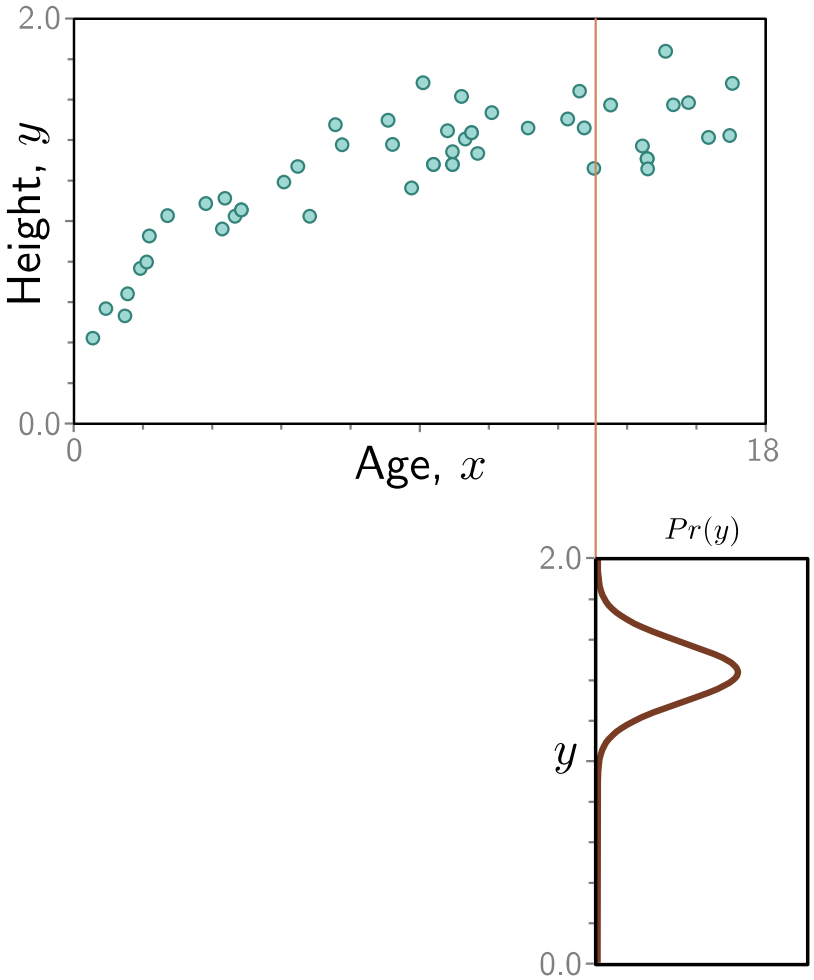
All models are wrong, but some are useful.
/George Box/


2.0
0.0
Height, y
0
18
Age, x
$Pr(y)$
2.0
y
0.0


2.0
0.0
Height, y
0
18
Age, x
$Pr(y)$
2.0
y
0.0

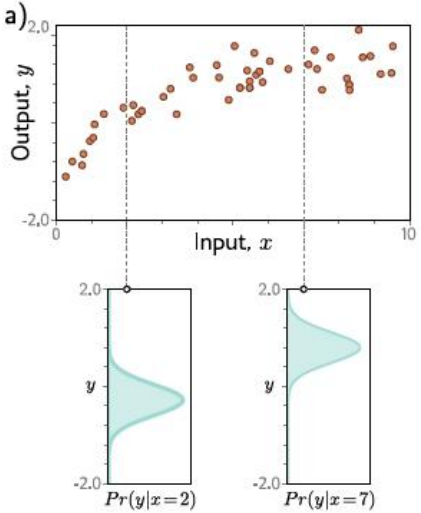

2.0
Height, y
0.0
0
18
Age, x
$Pr(y)$
2.0
y
0.0


2.0
0.0
Height, y
0
18
Age, x
$Pr(y)$
2.0
y
0.0

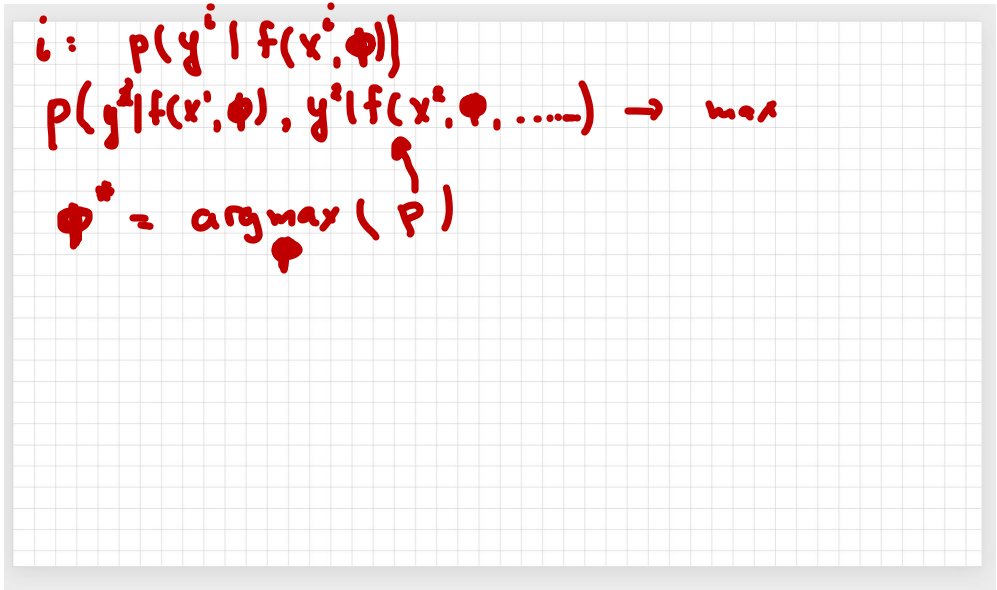

2.0
0.0
Height, y
0
18
Age, x
$Pr(y)$
2.0
y
0.0

Maximum likelihood

Consider a model $\boldsymbol{f} = x^{(i)}$, $\boldsymbol{\phi}$ parameterized with weights $\boldsymbol{\phi}$ that maps each i-th input sample $x^{(i)}$ into the output $z^{(i)}$ which then transforms into the hypothesis $h^{(i)}$ that should be close to the label $y^{(i)}$. We now shift perspective and consider the model as computing a conditional probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$ or $p(Y|X)$. The loss encourages each training output to have a high probability under the distribution $p\,y^{(i)}|x^{(i)}$.

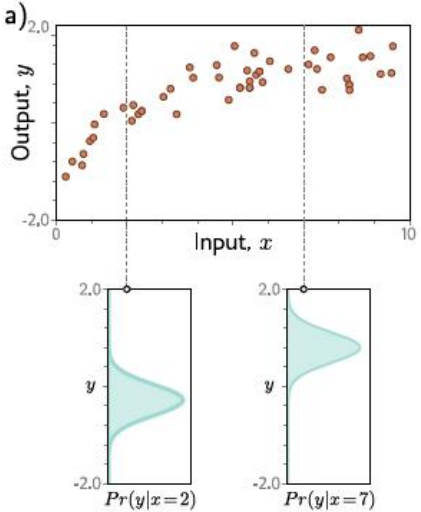


S. J. Prince. Understanding Deep Learning. MIT Press, 2023.



Maximum likelihood →minimum negative log-likelihood criterion

Two assumptions shold be met. First, we assume that the data are *identically* distributed. Second, we assume that the conditional distributions $p\left(y^{(i)}|x^{(i)}\right)$ are *independent.* In other words, the data are independent and identically distributed (i.i.d.).



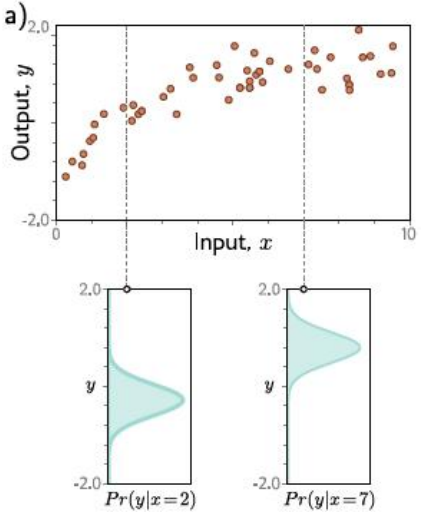
$$p(y^1|f(x^1,\Phi), y^2|f(x^2,\Phi)) = \prod_i p(y^i|f(x^i,\Phi))$$

$$\Phi^* = -\arg\min \sum \ln[p(y^i)|f(x^i;\Phi)] \Rightarrow \min$$

negative log-likelihood

S. J. Prince. Understanding Deep Learning. MIT Press, 2023.

Algorithm for constructing loss funtions



The recipe for constructing loss functions for training data $\{\mathbf{x}_i, \mathbf{y}_i\}$ using the maximum likelihood approach is hence:

1. Choose a suitable probability distribution $Pr(\mathbf{y}|\boldsymbol{\theta})$ defined over the domain of the predictions $\mathbf{y}$ with distribution parameters $\boldsymbol{\theta}$.
2. Set the machine learning model $\mathbf{f}[\mathbf{x}, \boldsymbol{\phi}]$ to predict one or more of these parameters, so $\boldsymbol{\theta} = \mathbf{f}[\mathbf{x}, \boldsymbol{\phi}]$ and $Pr(\mathbf{y}|\boldsymbol{\theta}) = Pr(\mathbf{y}|\mathbf{f}[\mathbf{x}, \boldsymbol{\phi}])$.
3. To train the model, find the network parameters $\hat{\boldsymbol{\phi}}$ that minimize the negative log-likelihood loss function over the training dataset pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\Big[L[\boldsymbol{\phi}]\Big] = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[-\sum_{i=1}^{I} \log\Big[Pr(\mathbf{y}_i|\mathbf{f}[\mathbf{x}_i, \boldsymbol{\phi}])\Big]\right]. \tag{5.6}$$

4. To perform inference for a new test example $\mathbf{x}$, return either the full distribution $Pr(\mathbf{y}|\mathbf{f}[\mathbf{x}, \hat{\boldsymbol{\phi}}])$ or the value where this distribution is maximized.

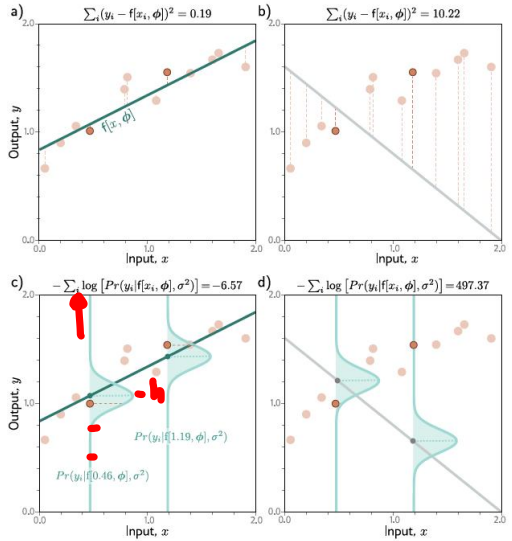
We devote most of the rest of this chapter to constructing loss functions for common prediction types using this recipe.

S. J. Prince. Understanding Deep Learning. MIT Press, 2023.

Case 1: Regression (univariate, homoscedastic) and MSE loss

The key idea is to use the *normal* distribution:

$$p(x) = N(x|\mu, \sigma^2)$$
$$= \frac{1}{\sqrt{2\pi\sigma^2}} exp - \frac{1}{2\sigma^2}(x-\mu)^2 .$$



$\sigma = const$

$$p^i(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{(y^i - h^i)^2}{2\sigma^2}\right)$$

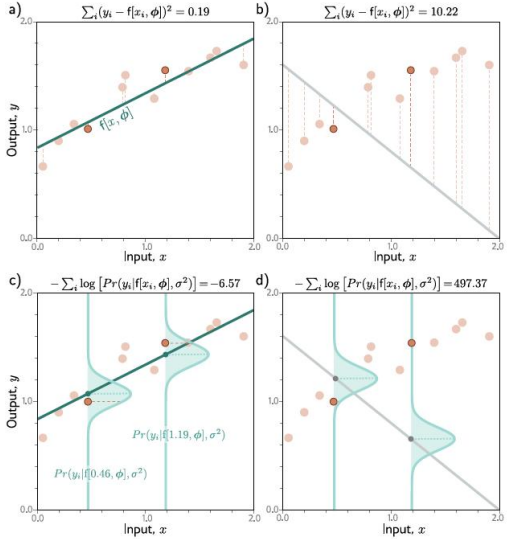
$$-\sum_i \ln(p^i(x)) = \ldots \Rightarrow min$$

S. J. Prince. Understanding Deep Learning. MIT Press, 2023.

Case 2: Regression (multivariate, heteroscedastic) and MSE loss

The key idea is to use the *normal* distribution:

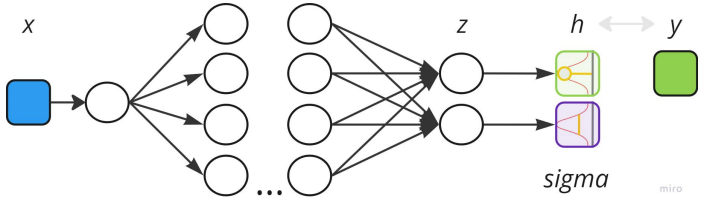
$$p(x) = N(x|\mu, \sigma^2)$$
$$= \frac{1}{\sqrt{2\pi\sigma^2}} exp - \frac{1}{2\sigma^2}(x-\mu)^2 .$$



S. J. Prince. Understanding Deep Learning. MIT Press, 2023.

An uncertainty-aware loss intuition

Homoscedastic regression *vs* heteroscedastic regression

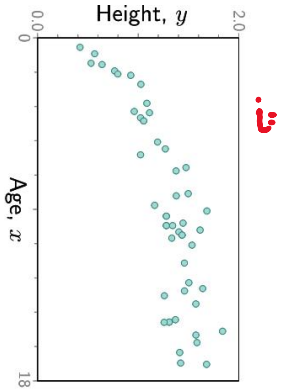


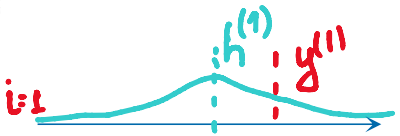
~~Model predicts output *h* given input *x*~~

Model predicts a conditional probability distribution *p(y|x)* over outputs *h* given input *x*

$$p\left(\mathbf{y}^{(i)}|\mathbf{f}[\mathbf{x}^{(i)},\mathbf{w}]\right) = \frac{1}{\sqrt{2\pi\sigma_{(i)}^2}} e^{-\frac{\left(y^{(i)}-h^{(i)}\right)^2}{2\sigma_{(i)}^2}}.$$

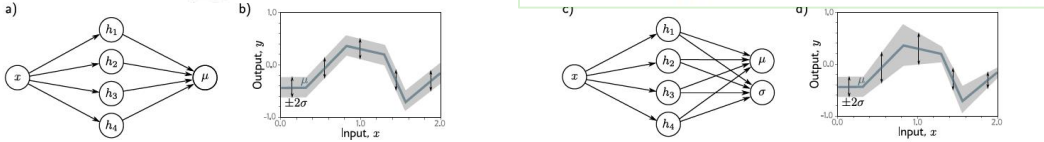
$$L = -\sum_{i=1}^{m} \log\left[\frac{1}{\sqrt{2\pi\sigma_{(i)}^2}} e^{-\frac{\left(y^{(i)}-h^{(i)}\right)^2}{2\sigma_{(i)}^2}}\right] \rightarrow min$$





$$L_R = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)}-h^{(i)}\right)^2$$

$$L_R = \frac{1}{2m}\sum_{i=1}^{m}\left(\frac{1}{\sigma_{(i)}^2}\left(y^{(i)}-h^{(i)}\right)^2 + \log\sigma_{(i)}^2\right)$$



Supervised learning intuition: S. J. Prince. Understanding Deep Learning. MIT Press, 2023. URL http://udlbook.com.

Questions



1. Why it is not recommended to differentiate functions obtained with approximation?
2. How can we estimate that *data* is enougth to solve a problem with an ML model?
3. Why small noise in *data* may be usefull?
4. Is it possible to make an AI model that solves all problems at once?
5. Is it possible to solve a problem using a complex *performance measure*? E.g. minimize something and maximize something else simultaneously?
6. How can the gradient descent method be improved to find global minima instead of local ones?
7. Can the discussed linear regression problems be solved analytically without using the gradient descent method?
8. Why is the use of high-degree polynomials generally not recommended when building regression models?

Thank you for your attention!

a.kornaev@innopolis.ru, @avkornaev



Notes