1 Introduction

Artificial Intelligence

- broad concept
- different interpretations
- we do not have a definition of inteligence

Statistical machine learning

- Algorithms and applications where computer learn from data

AGI

- Artificial General Intelligence
- Hypothetical computer program that can perform intellectual tasks as well as, or better than a human.

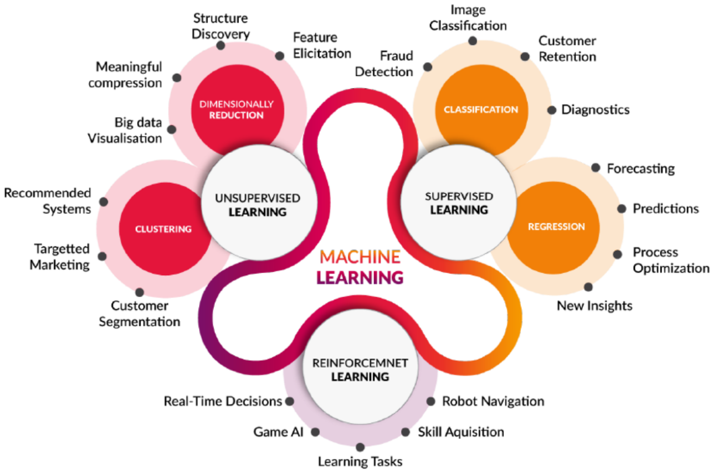
Turing Test

- Also called imitation game
- Tests of a machine's ability to exhibit intelligent behaviour equivalent to, or indistinguishable from that of a human
- Has some philosophical problems (Complex problems, humans cant solve / AI must learn to lie)

Examples of application (today):

- Personalization of news feeds
- Product searching and recommendation s on eCommerce platforms
- Voice-to-text
- Predictive maintenance

1.1 Tasks and Algorithms of Machine Learning



1.2 Natural Language Processing (NLP)

- Automated processing of human language (written & spoken)
- Aims to understand and generate human (natural) language
- Understanding spoken text is still difficult
- Understanding written text became BIG business (search-engines)
- Generating human-like conversations is still very hard

1.3 Dialogflow

Intents

- Recognizes the need of a user
- Require training to match to user inputs
- Follow up Intents (on Success)
- Fallback Intents (on Failure)

Entities

- Extract information from user inputs
- Help to identifiy required intent
- System Entities: (Date and time / Numbers / Amounts / Units / etc.)
- Developer Entities: defined by list of words (@pizza-type / @drink / etc.)
- User Entities: transient, temporary Information based on Conversation

Dialog

- Linear: Gather a list of information
- Non Linear: Using Contexts

Context

- Each Intent can have Input & Output Context
- Intents are active based on active Context
- Expire automatically

Fulfillment

- Action triggered on fullfilled Intents
- e.g. Webhook

2 Natural Language Processing (NLP)

2.1 Ingredients of Machine Learning

1. Data

- Dataset
- Pre-Processing Pipe-Line including cleansing, feature-engineering, data augmentation etc.

2. Cost-Function (Loss)

- Formal mathematical expression for good / bad
- Commonly Mean Squared Error (MSE)

3. Model

- From linear model: $\hat{y_i} = ax_i + b$
- To complicated million parameter neural networks
- Different tasks require different models (regression / decision tree)

4. Optimization Procedure

- Algorithm that changes the parameters of the model that the cost-function is minimized.
- E.g. Stochastic Gradient Descent (SGD), ADAM, RMSProp...

2.2 More ingredients

For successful ML, there are many more ingredients:

5. Performance optimization

- Building of efficient pipe-lines
- Folowwing tool specific recommendations

6. Visualization and evaluation of the learning Process

- Learning curves
- Performance measures
- Tensorboard

7. Cross-Validation & Regularization

- Train models that generalize well to unseen data
- Estimate the generalization error

2.3 Representation of Words

Vectors can be used to represent words based on their meaning.

2.3.1 One-hot representation

- Vector with a single 1-Value
- All other Values are set to 0
- Count the Number of different Words, Define one unique vector per word:

Dini Mom isch fett.

$$\text{Dini: } \begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix} \quad \text{Mom: } \begin{bmatrix}0\\1\\0\\0\\0\end{bmatrix} \quad \text{isch: } \begin{bmatrix}0\\0\\1\\0\\0\end{bmatrix} \quad \text{fett: } \begin{bmatrix}0\\0\\0\\1\\0\end{bmatrix} \quad \text{'.': } \begin{bmatrix}0\\0\\0\\0\\1\end{bmatrix}$$

Disadvantages:

- Very high dimensional vector space (1 Dimension / unique Word)
- Sparse Representation: Eech vector has a single 1 and N Zeroes. (Memory Inefficient)
- No Generalization: All words are unrelated to each other.
- Does not capture any aspect of the meaning of a word

2.3.2 Indexing

Make a list of words (optionally alphabetically). Use the index to represent each word.

Example:

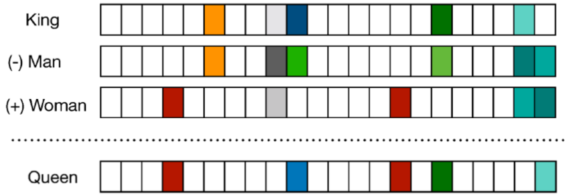
Dini Mom isch fett.

Dini: 0, Mom: 1, isch: 2, fett: 3, '.': 4

- Dense Equivalent of one-hot encoding
- Indexes are not more useful that one-hot vectors
- Often used as preprocessing step
- Indices / One-Hot Vectors are fed into a network which learns more useful representations

2.3.3 Distributed Representation

- Words that occur in similar contexts (neighboring words) tend to have similar meanings
- Similar words share similar representations
- Distributed representations can be learned



Words to Vectors:

- Mathematical function maps word to high dimensional Vector
- In neural networks, this function is implemented in the Embedding Layer

Advantage of Vectors

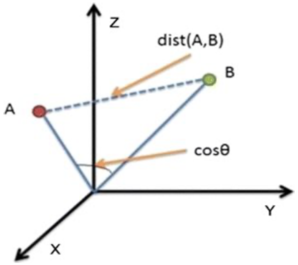
- Good embedding maps simiar/related words to similar regions of the vector space
- Dot-Product (Skalarprodukt) is a measure of similarity
- Possible to add/subtract vectors

Calculate Similarities between words Dot-Product (Skalarprodukt) of 2 Vectors:

- maximal when parallel (0°) (1 with norm (length) 1)
- zero when orthogonal (90°)
- minimal (negative) when opposite directions (180°) (-1 with norm (length) 1)

Cosine Distance

- Way to calculate how similar two words (vectors) are



$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

3 Probability

3.1 Random Variables

- Values depend on outcomes of a random phenomenon
- Random variable X is a variable that takes a numerical value x, which depends on a random experiment
- **Discrete:** X takes any of a finite set of values $1.5, 2.123, 6.2, 10$
- **Continous:** X takes any alue of an uncountable range e.g. real numbers from an interval

Best we can know

- All possible values
- Probability of each value

E.g. The discrete random variable X is the number observed when rolling a fair dice.

$Pr(X = x)$ / $P(x)$: 1/6 for each possible value

3.1.1 Two random variables

Joint Probability

- Joint Properties of two random variables
- Defined by the Joint Probability Mass Function

E.g. Dice1 = 5 AND Dice2 = 4

$P_{XY}(5, 4) = 1/36$

	X=1	X=2	X=3	X=4	X=5	X=6
Y=1	1/36	1/36	1/36	1/36	1/36	1/36
Y=2	1/36	1/36	1/36	1/36	1/36	1/36
Y=3	1/36	1/36	1/36	1/36	1/36	1/36
Y=4	1/36	1/36	1/36	1/36	1/36	1/36
Y=5	1/36	1/36	1/36	1/36	1/36	1/36
Y=6	1/36	1/36	1/36	1/36	1/36	1/36

Independant random Variables

- Joint Probability is the product of the individual probabilities

$P(X, Y) = P(X) * P(Y)$

$P(X, Y, Z) = P(X) * P(Y) * P(Z)$

Correlated random Variables

- There are events that are not independant
- Such random variables are correlated
- X: observe clouds (0=no, 1=small, 2=big)
- Y: observe rain (0=no, 1=light, 2=moderate, 3=heavy)

Conditional Probability

- One variable is no longer random
- X is observed, its value is fixed
- Calculate the probabilities of Y given X: $P(Y|X)$

$P(X, Y) = P(X|Y) * P(Y)$
$P(X, Y) = P(Y|X) * P(X)$
$P(Y|X) = \frac{P(X,Y)}{P(X)}$

Bayes Rule
$P(X|Y) * P(Y) = P(Y|X) * P(X)$
Therefore:
$P(Y|X) = \frac{P(X|Y)*P(Y)}{P(X)}$

4 Python

Chani alles

5 Data Visualization

- See trends, clusters and patterns in data
- Difficult to see in raw data
- Detect outliers and unusual groups
- Validate Hypothesis/Conjecture/Theory

Important in a Plot:

- X-Axis / Y-Axis
- Title
- Scale
- Dimensionality of the data 2D / 3D

5.1 Data Analysis Libraries

5.1.1 NumPy

- Package for scientific computing in Python
- Multidimensional array object
- Routines for fast array operations (sorting, selecting, FFT, linalg, etc)

5.1.2 pandas

- Built on top of NumPy
- Routines for accessing tabular data from files (.csv, xls, etc.)
- Supports 2-dimensional data (dataframe and series)
- Dataframes are something like database tables

5.1.3 MatPlotLib

- Library for visualizing data
- Bargraphs, Histograms, Piecharts, Scatter plots, lines, boxplots, heatmaps, etc.

5.1.4 Seaborn

- Extension of MatPlotLib, NumPy and pandas
- More user friendly
- Plots are aesthetically better

5.1.5 Chart types

Line Plots

- Bivariate, Continous
- Recognizes trend (pattern of change)

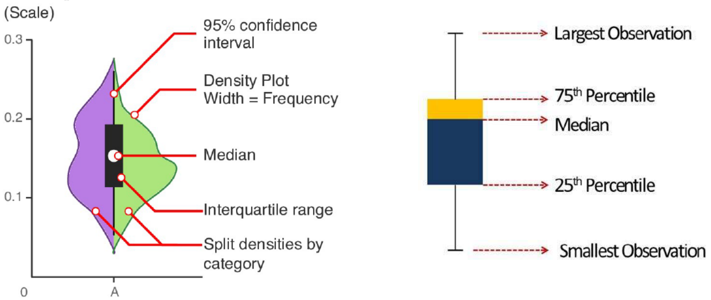
Bar Chart

- Used for categorical data
- Counting based on each category

Histogram

- Represents the empirical distribution of a variable
- Automatically creates bins (interval) along the range of values
- Shows vertical bars to indicate the number of observations / bin

Descriptive Statisics: Box Plots and Violin Plots



Scatter Plot

- Relationship between continous variables
- Helps to get an idea of the degree of correlation between variables

6 Regression

6.1 What is a model?

In ML, we use the term **model** for any mathematical function that explains the data:
$y_i = f(x_i)$
$y_i = f(x_i) + \epsilon_i$

where ϵ_i is unexplained noise. It is often assumed that ϵ_i follows a normal distribution.
Instead of approximating y_i, we calculate an **estimate** $\hat{y_i}$ (y hat) of the usually unknown y_i:

$$\hat{y_i} = f(x)$$

6.1.1 Linear Regression

- Only consideres a linear relationship between input and output
- In the simplest case, x and y are scalars and the linear model therefore has only two free parameters
- The goal is to identify a (slope) and b (intercept) for which the linear model best explains the data

$$\hat{y_i} = ax_i + b$$

6.1.2 Mean Squared Error (MSE)

- Loss we want to minimize
- Usually divided by 2

$$\hat{y_i} = ax_i + b$$
$$e_i = y_i - \hat{y_i}$$
The difference e_i, called residual
$$E = \frac{1}{2N} * \sum_{i=1}^{N} e_i^2$$
$$E = \frac{1}{2N} * \sum_{i=1}^{N} (\hat{y_i} - (a * x_i + b))^2$$

6.1.3 Correlation and Causality

- Correlation is not causality
- Correlation refers to the degree to which a pair of variables are linearly related
- Linear regression is a tool to detect correlations between two or more variables
- Correlation can be quantified using the Pearson correlation coefficient