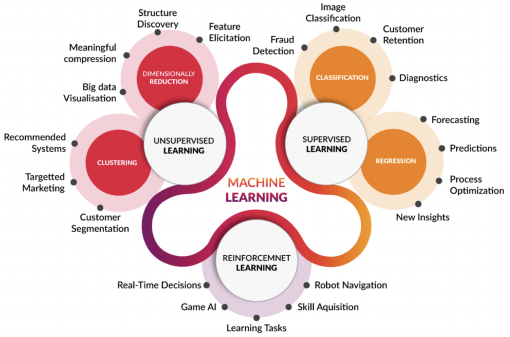
1 What is Artificial Intelligence?



Supervised Learning: System training with features and labels Examples: Visual Recognition, Email Spam Detection, Fraud Detection
Unsupervised Learning: System training without labels Example: Social network analysis, Recommendation Systems, Astronomical data, clustering of news articles.
Reinforced Learning: A system learn based on negative or positive feedback to do the right things. Example: AI for Arcade Game.

1.1 Turing Test (Imitation Game)

Is a test of a machine's ability to exhibit intelligent behaviour equivalent to, or indistinguishable from, that of a human. A Person is in a room in front of two screens and has a conversation with a human and a computer and can clearly say which one is the computer.

2 Dialogflow

3 The four Ingredients of ML

Data The dataset we are given plus the pre-processing pipe-line including cleansing, feature-engineering, data-augmentation etc.
Cost-Function (Loss) A formal mathematical expression for good and bad (Mean Square Error (MSE), etc.).
Model Something simple as the linear model $\hat{y}_i = ax_i + b$ or a million-parameter neural network. Different tasks require different models. → domain-knowledge required to find best model
Optimization Procedure Algorithm that changes the parameters of the model such that the cost-function is minimized (SGD, ADAM, RMSProp).
Performance Optimization Building efficient pipe-lines is difficult.
Visualization and Evaluation of learning process Learning curves, Performance measures.
Cross-Validation & Regularization Goal: train models that generalize well to unseen data.

4 Naturale Language Processing

4.1 One-hot Vector

A One-hot vector is a vector with a single value 1 and all other set to 0. We create for each word a unique vector. **Disadvantages:** High dimensional, sparse representation, no generalisation (all words are unrelated to each other), does not capture the meaning.

4.2 Indexing

An index is used for every word. Indexing is the dense equivalent of One-hot encoding and indexes are more useful than vectors. **Disadvantages:** Indexing is often a preprocessing step and this indices are then fed into a network to learn representations (embeddings).

4.3 Distributed Representation (dense vectors)

Distributed representations are the opposite of One-hot, instead of concentrating the meaning of a data point into one component or one "element", the meaning of the data is distributed across the whole vector. **Finding D.R.** is a difficult task, needs a lot of time, data and CPU/GPU. In practice: download a predefined language model, use an Embedding Layer and optimize the embedding for the task. → Similar words share similar represenations. DR can be learned **Predefined Word Embedings:** GloVe, Word2Vec.

4.4 Word to Vector

A mathematical function maps input into output. In neural networks this function is implemented in a called Embedding Layer.

4.5 Calculate Similarity

We can calculate the similarity of two words as vectors with the cosine distance.

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

5 Discrete Random Variables

There are two types of random variables. **Discrete RV:** X takes any of finite set of values, e.g. 1.5, 2, 4. **Continues RV:** X takes any value of a range, e.g. $[2; 7]$

5.1 Probability Mass Function (PMF)

Is a function $f(X)$ that provides the probability for each value x of a discrete random variable X. A PMF specifies a discrete Probability Distribution.

5.2 Joint, Conditional Probability (JPMF)

The join property of two random variables are defined by the Joint Probability Mass Function. $PR(X = 5, Y = 4) = \frac{1}{36}$ **Independent events can be multiplied.**
Conditional Probability: $Pr(Y|X)$. Joint Prob. $P(X, Y) = P(X|Y)P(Y)$

5.3 Bayes Rule

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

6 Data Visualization

Data visualization means drawing graphic displays to show data and is generally the first step of machine learning. It gives an intuitive understanding of data structure, helps to see trends, clusters, patterns which we don't see in raw data.

6.1 Plot Types

Components: X & Y-Axis Labels, Title, Scale (linear vs logarithmic), Dimensionallity.

6.1.1 Line Plots

The line connects several distinct data points, presenting them as one continuous evolution. The result is a simple, straightforward way to visualize changes in one value relative to another.

6.1.2 Bar Chart

Often used for categorical data where binning/counting is done based on each category.

6.1.3 Histogramm

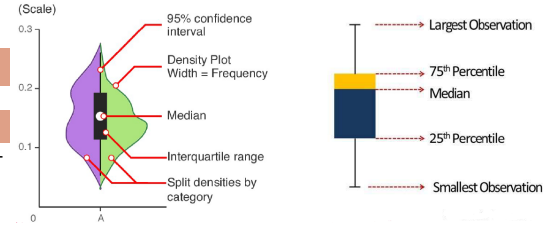
A histogram represents the empirical distribution of a variable by automatically creating bins (interval) along the range of values and by showing vertical bars to indicate the number of observation in each bin.

6.1.4 Descriptive Statistics

Image to the right.

6.1.5 Scatter Plot

To understand the relationship between two continuous variables a scatterplot can be used.



7 Linear Regression (Supervised Learning)

Linear Regression is the most used method to analyse data. In linear regression, we decide to only consider a linear relationship between the input and output. Most of the applications fall into two categories.
Interpretation: We want to understand if some input has an effect on the output. **Example:** Is there a relationship between smoking cigaretts and the risk of lung cancer?
Prediction: Given some sensor data like oil pressure, temperature etc. a model could predict an engine failure.

7.1 Machine-Learning Perspective

We are given both the input and the labels. The ML algorithm learns the linear relationship between X and Y. The goal is to predict a y_i for some unseen x_i. We use a model to explain the data. $y_i \approx f(x_i) + \varepsilon_i$ where the ε is unexplained noise.

The Simplest model is the linear model with two free (unknown parameters). $\hat{y} = ax_i + b$ where a=slope, b=intercept.

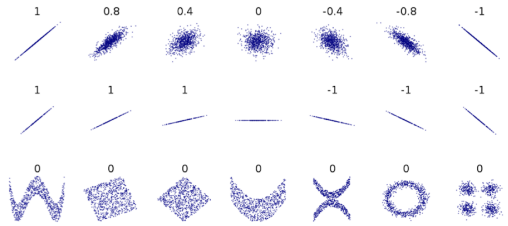
7.2 Mean Squared Error (MSE)

Is the loss we want to minimise in the ML model. $\hat{y}_i = a * x_i + b$ with the **Residual** $\varepsilon_i = y_i - \hat{y}_i$ Note: The sum of the squared residuals is the loss function. $E = \frac{1}{2N}\sum_{i=1}^{N} e_i^2$

7.3 Correlations

Correlation is not causality! Correlation refers to the degree to which a pair of variables are linearly related. This can be quantified with the Pearson correlation Coefficient (cov=covariant, σ=standard deviatoin).

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$



8 Gradient Descent

Gradient Descent is an iterative method. At each iteration, the model parameters are updated such that the Loss (MSE) is reduced. Other Loss function than the MSE can be choosen. The GD and SGD is very sensitive to the learning rate. **Problem:** For big Data the GD is very slow.

1. Take the derivative of the Loss Function for each parameter in it. (Gradient of Loss Function)
2. Pick random values for parameters
3. Plug the parameter values int the derivates (Gradient)
4. Calculate the Step Size= $Slope * LearningRate$.
5. Calc new Parameters = $OldParameter - StepSize$
6. if(!(step size $>$ x || max steps taken)) go to step 3.

8.1 Stochastic Gradient Descent (SGD)

At each iteration step, the gradient is calculated on a (randomly chosen) subset of the data (strict definition = one sample per step). Most of the time a mini-batch of data is selected. **(simulated) annealing:** learning rate is reduced over time. If new data is added we don't have to do all calculations again. There are many differnet options (called schedules) how to reduce alpha over time. Here we apply an exponential decay. Gradient Descent is the basic building block behind many variants: Adam, Adagrad, RMSProp etc.

9 Generalization and Regulation

In-sample error: It is possible to find a model which perfectly fits the data (MSE = 0) This is the loss minimized during the training phase, it is therefore also called **training error**.
→ Our goal is to learn a model from data that generalizes well to new data. A good model has a low generalization error.
Test Error (Generalization Error): We get this by using two completely disjoint datasets: one to train the model and the other to calculate the classification error. Both datasets need to have values for y. The first dataset is called training data and the second, test data.

9.1 Underfitting (High Bias)

When we have the scenarion of underfitting we have choosen a too simple model. (in-sample error=Large, Generalization error=Larger)

9.2 Overfitting (Low Bias)

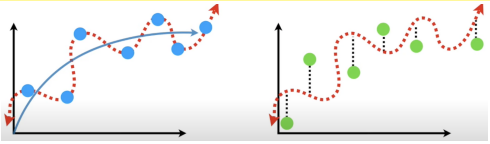
We have the scenarion of overfitting if we have choosen a too complex model (in-sample error=0, Generalization error=huge).

9.3 Trainig-Set, Test-Set

We cant calculate the generalization error! We can try to estimate the generalization error. We can split the data into a training-set and test-set (80/20). We fit the model to the training set and minimize the in-sample error. We evaluate the model with the test-set which estimates the generalization error.

9.4 Bias Variance Tradeoff

Bias: The inability for a ML method to capture the true relationsship (High match=low Bias, low match=high Bias)
Variance: is the difference in fits between two data sets. Higher bias implies lower variance, lower bias implies higher variance



Low Bias, High Variance

Note that we use the expression high bias in the sense of a too simple model for the given data and vica versa. High bias is a design by choice. → Find an optimal balance between bias and variance.

9.5 Generalization

We have to find the optimal balance between the variance and the bias with the help of generalization. We need to ingridient for regularization.

1. A Way to measure complexity: L1-Norm, L2-Norm
2. A Way to control model complexity: penalty to loss, complex models get higher penalty.

9.6 Regularization

Regularization is the most used technique to penalize complex models in machine learning, it is deployed for reducing overfitting by putting network weights small. Also, it enhances the performance of models for new inputs.

9.6.1 L1-Norm (LASSO)

It adds an L1 penalty that is equal to the absolute value of the magnitude of coefficient, or simply restricting the size of coefficients. λ= Regularization Parameter (Hyper Par), β=Polynomial arguments. Lasso regression can do feature selection if a β_i=0.

$$Cost = \frac{1}{2N}\sum_{i=1}^{N}(\hat{Y} - Y)^2 + \lambda\sum_{i=1}^{N}|\beta_i|$$

9.6.2 L2-Norm (Ridge Regression)

It adds an L2 penalty which is equal to the square of the magnitude of coefficients. β_0 is not regularized.

$$Cost = \frac{1}{2N}\sum_{i=1}^{N}(\hat{Y} - Y)^2 + \lambda\sum_{i=1}^{N}\beta_i^2$$

9.6.3 L1 + L2 Norm (Elastic Net)

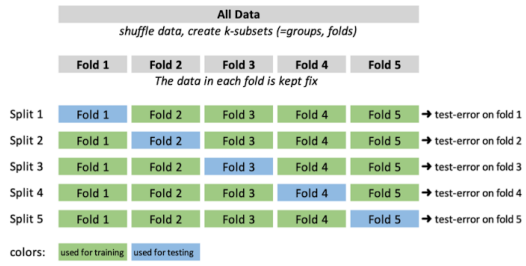
Elastic net uses both the L2 and the L1 penalty.

10 Cross Validation

With CV we want to solve a problem we introduce with the split in training and test data. One problem with this approach is the following: If we select 20% (or some other fraction) for testing, then the test-error depends on a (potentially small) random set. For a different set, we could get a different test-error. → CV is a technique to compare (and select from) different models (=different parameter values) **Use Cases:** estimation of generalization error, selection of hyper parameters.

10.1 k-Cross Validation

We can apply cv to obtain a better estimate of the generalization error.
In k-fold cross-validation, we repeat the split-train-test procedure k times, using a systematic resampling procedure.
The data is split once into k folds. Then train/test is repeated k-times. Each fold participates in k-1 training phases and is used once for testing:



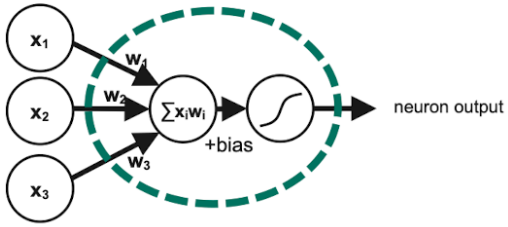
10.2 Selection of Hyper Parameters

The split/train pattern of cross validation can be used to find optimal hyper parameters.

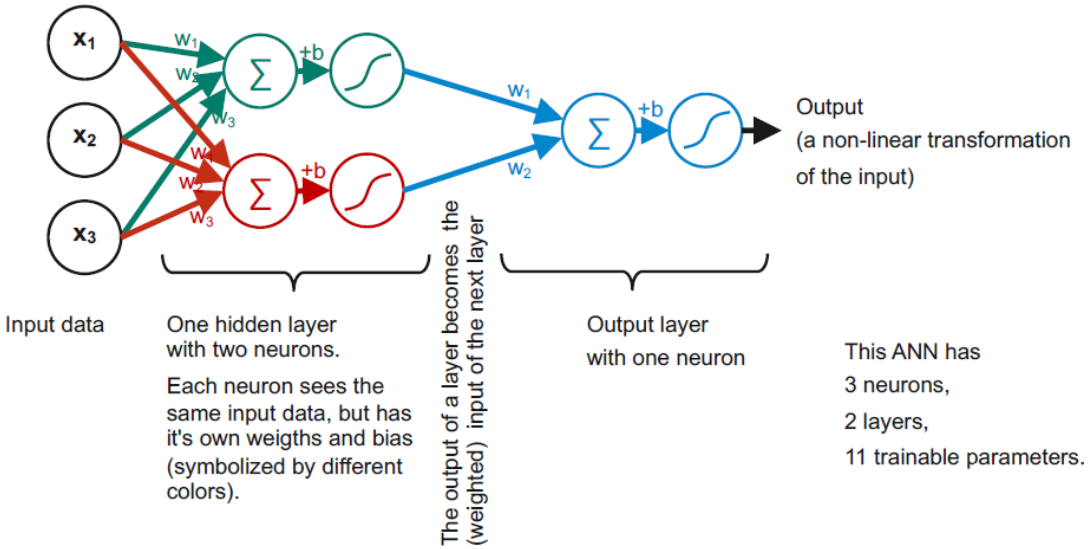
11 Artificial Neural Networks

11.1 Neuron

The artificial neuron receives an input vector $[x_1, x_2, x_3, ...]$. Each neuron has its own input weights $[w_1, w_2, w_3, ...]$ and bias $b(= intercept)$. The neuron calculates the sum of the weighted input (dot product $\vec{x} * \vec{w}$), adds a bias b, and passes it through a nonlinear activation function (=only positive values pass).



11.2 Artificial Neuron Network (ANN)



11.3 How to train a ANN

We consider a simple case of supervised learning: for each input we are given the output (target values/labels). The ANN is initialized with random weights and produces some output $\hat{y}$. Then, an optimizer (e.g. SGD) reduces a cost-function (e.g. MSE).

That is, at every iteration, and for every single weight w (& and every bias b), the partial derivative $\frac{\partial}{\partial w}(y - \hat{y})^2$ needs to be calculated. Luckily there's an algorithm which is doing this very efficiently: **Backpropagation**.

12 Binary Classification and Linear Regression

A binary classification problem has only two possible outcomes: yes, no. The algorithm for solving binary classification is logistic regression.
Optimize the model using the probabilities and not the response. In other words: We want a cost function over probability of classmembership y.

12.1 Binary Classification

Why not with linear regression? LR optimizes the wrong quantity.
Problems: Using LR, we model the response (y) and post process the response (e.g. by thresholding) to compute the probability. → The cost function minimizes the MSE difference in the values of $\hat{y}$ and y and has nothing to do with the clasification probabilities.

12.2 Logistic Regression

The sigmoid function maps all input to a probability. This is where our features (data x) enters the calculation. The weights are unknown and need to be learned. $y = w_1x_1 + w_2x_2...$

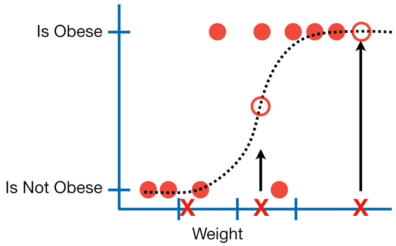
Can work with continous data and discrete data.

$$sigmoid(y) = \frac{1}{1 + e^{-y}}$$

The weights is a model parameter.
$Pr(y = 1|(featurevector); (Weights))$

$$Pr(y = 1|x) = Pr(y = 1|x) = p(x) = \frac{1}{1 + e^{-(W^T x_i)}}$$

W is a vector can be found with Maximum Likelihood.



12.3 Maximum Likelihood

Given all the data points (X, Y) we want to maximise the probability that all the predictions are correct. The objective off the training is to set the coefficients W so that: p is close to 1 when y=1 and p is close to 0 when y=0. To find the perfect W we can use Gradient Descent.

$$MaximumCost_2(W) = \sum_{y=1}^{N} \log(p(x_i) + \sum_{y=0}^{N} \log(1 - p(x_i))$$

13 Classifier Evaluation

For the evaluation of a classifier we have different measurements.

13.1 Confusion Matrix

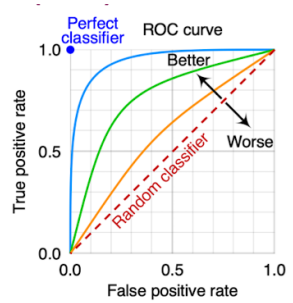
False Positive: Model=yes, Truth=no
False Negative: Model=no, Truth=yes
True Positive: Model=yes, Truth=yes
True Negative: Model=no, Truth=no
Accuracy is not enough! Consider an in-balanced Dataset where 10% is false and 90% true. Increasing precision reduces recall and vice versa.

Mean Accuracy:$= (T_p + T_n/n)$
Mean Error:$= (F_p + F_n/n)$
Precision:$= T_p/(T_p + F_p)$
Recall, True Positive Rate (TPR):$= T_p/(T_p + F_n)$
Miss Rate, False Negative (FNR):$=$ 1 - TPR

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP),	False negative (FN),
	Negative (N)	False positive (FP),	True negative (TN),

13.2 Receiver Operating Characteristics (ROC)

A ROC space is defined by FPR and TPR as x and y axes, respectively, which depicts relative trade-offs between true positives (benefits) and false positives (costs).
Area under the curve is the area under the ROC. The greater the area under the curve, shows the higher quality of the model. The greater the area under the curve, the higher the ratio of true positives to false positives.



14 KNN Classification (Unsupervised Learning)

14.1 Linear Separability, Decision Boundary

Two subsets are said to be linearly separable if there exists a hyperplane that separates the elements of each set in a way that all elements of one set resides on the opposite side of the hyperplane from the other set. In 2D plotting, we can depict this through a separation line, and in 3D plotting through

a hyperplane. If we have additional information about the structure of the data, we can apply a non-linear transformation.

14.2 Logistic Regression

- Parametric model (W) → Needs training to find optimum W
- Computes the probabilities (p), not the class
- A suitable threshold needs to be determined (ROC/AUC, FPR vs TPR) to decide the class
- Multiple classes can be supported recursively (slow and can result in ambiguity)
- Linear decision boundaries between classes

14.3 k-Nearest Neighbors

A datapoint is known by the company it keeps. It has two Hyperparameters: k value, distance function. **Advantages:** Easy and simple machine learning model, Few hyper parameters to tune. **Disadvantages:** k should be wisely selected, Large computation cost during runtime if sample size is large, Not efficient for high dimensional datasets.

14.3.1 Distance Metric

There are three different metrics which can be used in the KNN algorithm. We have the distannce $x_1 = (x_{1,1}, x_{1,2}, x_{1,n})$ and $x_2 = (x_{2,1}, x_{2,2}, x_{2,n})$

Cosine: (Skalar)

$$\frac{x_1 * x_2}{||x_1|| * ||x_2||}$$

Manhattan: (Quadrat)

$$\sum_{i=1}^{n} |x_{1,n} - x_{2,n}|$$

Euclidian: (Kreis)

$$\sqrt{\sum_{i=1}^{n} (x_i - x_i)^2}$$

15 Clustering (Unsupervised Learning)

As the examples are unlabeled, clustering relies on unsupervised machine learning. If the examples are labeled, then clustering becomes classification. The goal of unsupervised learning is to self-discover patterns from the data. Used for social network analysis, astronomical data, google news articles for category, marked segmentation, recommendation systems

15.1 k-Means Clustering

Initialization: performance depends on the random initialization of the seeds for the centroids (Initialize randomly, run multiple times).
Stopping Criterion: When centres don't change. The datapoints assigned to specific cluster remain the same. The distance of datapoints from their centre (Threshold). Fixed number of iterations have reached.
Standardization of Data: Features with large values may dominate the distance value. Features with small values will have no impact in the clustering. We should normalise values.

1. We define the number of clusters k_c
2. Initialize the value of k cluster centres (centroids) $C_1.C_2..C_{k_c}$
3. Assignment
   - Find the squared Euclidean distance between the centres and all the data points
   - Assign each data point to the cluster of the nearest centre.
4. Update: Update the center for each cluster.
5. If (stoppin criterion met) { Stop } else {go to assignment}

Caluclate Cluster Center: $C_{R_x} = \frac{Sumof-x-coefficients}{datapoints}$, $C_{R_y} = \frac{Sumof-y-coefficients}{datapoints}$

15.2 Evaluation: Within-cluster sum-of-squares (WCSS)

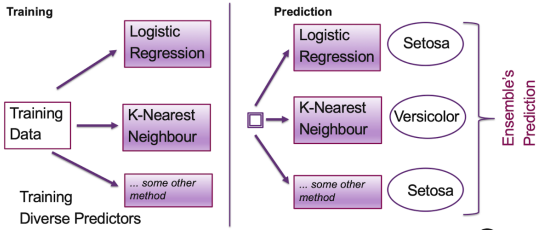
Sum of squared distances of samples to their closest cluster centre.

15.3 Evaluation: Silhouette Score

How far away the datapoints in one cluster are, from the datapoints in another cluster. Shilloute Score for point a $= \frac{b-a}{max(a,b)}$ where a=average intra-cluster distance i.e the average distance between each point within a cluster, b=average inter-cluster distance i.e the average distance between a cluster and its nearest neighbour cluster.

16 Ensemble Methods

Instead of expecting a single model to perform best on all data, we train multiple models and aggregate their results. Techniques: Bagging, Pasting.



16.1 Dimensions

We can use multiple methods for the classification of our data. These different methods can then be combined. Different learners use different algorithms, hyperparameters and training data. Additionally we can also apply the Cross Validation Pattern.

16.2 Voting

We can aggregate the predictions of weak learners with either soft or hard voting.
Hard Voting: Predict the class that gets the most votes
Softvoting: Predict the class with the highest class probability, averaged over all classifiers.

16.3 Bagging and Pasting

Sampling with replacement (Bagging): a data point can be selected more than once
sampling without replacement (Pasting): a data point can be selected only once.

16.3.1 Out of Bag evaluation

In Bagging, some data points may be used several times.

16.4 No free lunch theorem

No single machine learning algorithm is universally the best-performing algorithm for all problems. Evaluations! Since a predictor never sees out-of-bag (oob) data, it can be evaluated on oob data. No need for a separate validation or cross-validation set.