Nature–inspired and deep methods for feature selection

Pavel Krömer Jan Platoš[1]

Data Science Summer School @ Uni Vienna

[1]Dept. of Computer Science,
VŠB - Technical University of Ostrava,
Ostrava, Czech Republic
{pavel.kromer,jan.platos}@vsb.cz

Outline

Introduction

Introduction

Problem statement

Modern datasets comprise of millions of records, many thousands of features.

Introduction

Problem statement

Modern datasets comprise of millions of records, many thousands of features.

Feature (subset) selection is an established procedure to reduce data dimensionality, which is good for performance and accuracy (of e.g. classification).

Introduction

Problem statement

Modern datasets comprise of millions of records, many thousands of features.

Feature (subset) selection is an established procedure to reduce data dimensionality, which is good for performance and accuracy (of e.g. classification).

Nature–inspired feature selection methods, based on the principles of evolutionary computation, have shown potential to efficiently process very-high-dimensional datasets.

Feature subset selection

Feature subset selection (FSS) is a high–level search for an optimum subset of data features selected according to a particular set of criteria.

Feature subset selection

Feature subset selection (FSS) is a high–level search for an optimum subset of data features selected according to a particular set of criteria.

In a data set, $Y = \{A \cup Z\}$, $A = \{a_1, a_2, \ldots a_n\}$ is a set of input features, find $B \subset A$ so that $f_{\text{eval}}(B)$ is maximized.

Feature subset selection

Feature subset selection (FSS) is a high–level search for an optimum subset of data features selected according to a particular set of criteria.

In a data set, $Y = \{A \cup Z\}$, $A = \{a_1, a_2, \ldots a_n\}$ is a set of input features, find $B \subset A$ so that $f_{\text{eval}}(B)$ is maximized.

FSS can be formulated as an optimization or e.g. search problem.

Feature subset selection

Feature subset selection (FSS) is a high–level search for an optimum subset of data features selected according to a particular set of criteria.

In a data set, $Y = \{A \cup Z\}$, $A = \{a_1, a_2, \ldots a_n\}$ is a set of input features, find $B \subset A$ so that $f_{\text{eval}}(B)$ is maximized.

FSS can be formulated as an optimization or e.g. search problem.

The definition of the evaluation criteria is a paramount aspect of evolutionary feature selection that highly depends on the purpose of the FSS.

Nature–inspired feature subset selection

Evolutionary computation

Evolutionary computation is a group of iterative stochastic search and optimization methods based on the programmatical emulation of successful optimization strategies observed in nature. Evolutionary algorithms use Darwinian evolution and Mendelian inheritance to model the survival of the fittest using the processes of selection and heredity.

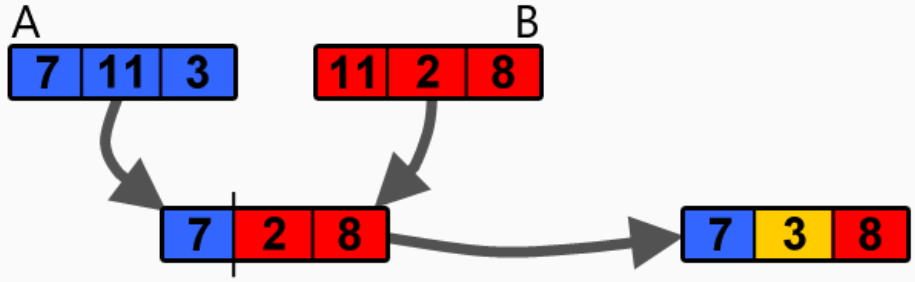
Genetic algorithms

The Genetic Algorithm (GA) is a population-based, meta-heuristic, soft optimization method. GAs can solve complex optimization problems by evolving a population of encoded candidate solutions. The solutions are ranked using a problem specific fitness function. Artificial evolution, implemented by iterative application of genetic and selection operators, leads to the discovery of solutions with above-average fitness.

Basic principles of GA

Encoding

Problem encoding is an important part of GA. It translates candidate solutions from the problem domain (phenotype) to the encoded search space (genotype) of the algorithm. The representation specifies the chromosome data structure and the decoding function.



Genetic operators

Crossover recombines two or more chromosomes. It propagates so called building blocks (solution patterns with above average fitness) from one generation to another, and creates new, better performing, building blocks.

In contrast, mutation is expected to insert new material into the population by random perturbation of chromosome structure. This way, new building blocks can be created or old disrupted.

Differential evolution

Differential evolution (DE) is a versatile stochastic evolutionary optimization algorithm for real-valued problems. It uses differential mutation

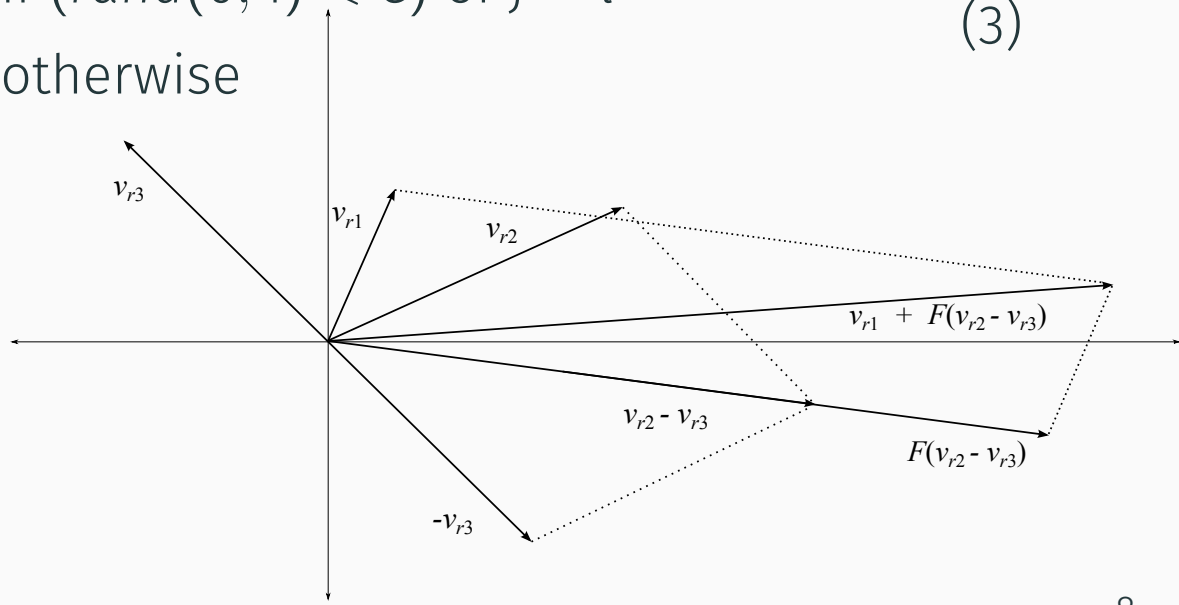
$$\vec{v}^{\,i} = \vec{v}^{\,r1} + F\left(\vec{v}^{\,r2} - \vec{v}^{\,r3}\right), \tag{1}$$

and crossover operator

$$l = rand(1, N), \tag{2}$$

$$v_j^i = \begin{cases} v_j^i, & \text{if } (rand(0,1) < C) \text{ or } j = l \\ x_j^i, & \text{otherwise} \end{cases} \tag{3}$$

to evolve a population of parameter vectors.



Evolutionary feature subset selection

Evolutionary FSS Types

Wrapper–based approaches look for subsets of features for which particular classification algorithm reaches the highest accuracy.

Evolutionary feature subset selection

Evolutionary FSS Types

Wrapper–based approaches look for subsets of features for which particular classification algorithm reaches the highest accuracy.

Filter–based approaches are classifier independent and utilize various indirect feature subset evaluation measures (e.g. statistical, geometric, information-theoretic).

Evolutionary feature subset selection

Evolutionary FSS Types

Wrapper–based approaches look for subsets of features for which particular classification algorithm reaches the highest accuracy.

Filter–based approaches are classifier independent and utilize various indirect feature subset evaluation measures (e.g. statistical, geometric, information-theoretic).

Here, we use two evolutionary methods for fixed–length subset selection and a fitness function based on compression–based data entropy estimation to establish a novell filter–based evolutionary FSS.

Compression–based data entropy estimation

Entropy is a general concept that expresses the amount of information contained in a message.

Compression–based data entropy estimation

Entropy is a general concept that expresses the amount of information contained in a message.

Entropy of a random variable, X, consisting of a sequence of values, $x_1, x_2, \ldots, x_n$, is defined by

$$H(X) = -\sum_i P(x_i) \log_2 P(x_i) \tag{4}$$

Compression–based data entropy estimation

Entropy is a general concept that expresses the amount of information contained in a message.

Entropy of a random variable, X, consisting of a sequence of values, $x_1, x_2, \ldots, x_n$, is defined by

$$H(X) = -\sum_i P(x_i) \log_2 P(x_i) \tag{4}$$

Entropy is used as a basis of a number of derived measures including conditional entropy, $H(X|Y)$, and information gain.

Compression–based data entropy estimation

Entropy is a general concept that expresses the amount of information contained in a message.

Entropy of a random variable, X, consisting of a sequence of values, $x_1, x_2, \ldots, x_n$, is defined by

$$H(X) = -\sum_i P(x_i) \log_2 P(x_i) \tag{4}$$

Entropy is used as a basis of a number of derived measures including conditional entropy, $H(X|Y)$, and information gain.

It is the basis of several feature selection methods, but is generally hard to evaluate in practical settings.

Compression–based data entropy estimation

Entropy is a general concept that expresses the amount of information contained in a message.

Entropy of a random variable, X, consisting of a sequence of values, $x_1, x_2, \ldots, x_n$, is defined by

$$H(X) = -\sum_i P(x_i) \log_2 P(x_i) \tag{4}$$

Entropy is used as a basis of a number of derived measures including conditional entropy, $H(X|Y)$, and information gain.

It is the basis of several feature selection methods, but is generally hard to evaluate in practical settings.

Computationally efficient entropy estimators are used in place of exact measures.

Compression–based data entropy estimation (cont.)

Compression–based data entropy estimation

A computationally feasible approach to entropy estimation for real–world applications with solid theoretical background (Shannon Entropy $\approx$ Kolmogorov complexity).

Compression–based data entropy estimation (cont.)

Compression–based data entropy estimation

A computationally feasible approach to entropy estimation for real–world applications with solid theoretical background (Shannon Entropy $\approx$ Kolmogorov complexity).

Kolmogorov complexity (of a binary string), $K(x)$, is the length of the shortest program that can produce x.

Compression–based data entropy estimation (cont.)

Compression–based data entropy estimation

A computationally feasible approach to entropy estimation for real–world applications with solid theoretical background (Shannon Entropy $\approx$ Kolmogorov complexity).

Kolmogorov complexity (of a binary string), $K(x)$, is the length of the shortest program that can produce x.

Conditional Kolmogorov complexity, $K(x|y)$, is analogous to conditional entropy.

Compression–based data entropy estimation (cont.)

Compression–based data entropy estimation

A computationally feasible approach to entropy estimation for real–world applications with solid theoretical background (Shannon Entropy $\approx$ Kolmogorov complexity).

Kolmogorov complexity (of a binary string), $K(x)$, is the length of the shortest program that can produce x.

Conditional Kolmogorov complexity, $K(x|y)$, is analogous to conditional entropy.

Kolmogorov complexity is non–computable, but has been associated with data compression (Li et al., 2004; Cilibrasi and Vitanyi, 2005).

$$K(x|y) \approx C(x \cdot y), \tag{5}$$

given $C(x) \approx C(x \cdot x)$.

Compression–based evolutionary feature subset selection

Objective

Develop a filter–based evolutionary feature subset method with entropy (compression) as the basis for feature subset evaluation (i.e. solve a specific fixed–length subset selection problem).

Compression–based evolutionary feature subset selection

Objective

Develop a filter–based evolutionary feature subset method with entropy (compression) as the basis for feature subset evaluation (i.e. solve a specific fixed–length subset selection problem).

Methods

Genetic algorithms (GA) – a GA for fixed–length subset selection with compact chromosomes, crossover and mutation, w/o creation of invalid individuals.

Differential evolution (DE) – a no–frills DE for fixed–length subset selection to see how a continuous algorithm does.

FPC, a fast lossless compression algorithm for double-precision floating-point data (Burtscher and Ratanaworabhan, 2009) as the fitness function.

Experiments

An in-house implementation of GA (steady-state, with generation gap 2) and DE (/DE/rand/1) with FPC as fitness function.

Experiments

An in-house implementation of GA (steady-state, with generation gap 2) and DE (/DE/rand/1) with FPC as fitness function.

Two data sets from the UCI Machine Learning Repository (Hepatitis, Spambase)

Experiments

An in-house implementation of GA (steady-state, with generation gap 2) and DE (/DE/rand/1) with FPC as fitness function.

Two data sets from the UCI Machine Learning Repository (Hepatitis, Spambase)

A battery of well-known classification methods (CART, Naive Bayes, k-Nearest Neighbours)

Data set properties and the number of classification errors for full data sets.

			Classification errors			
Dataset	Attrs.	Records	CART	NB	kNN(1)	kNN(3)
Hepatitis	20	80	0	11	8	13
Spambase	58	4601	3	513	3	216

Experiments (cont.)

FPC as a feature subset selection criterion

All possible subsets of 2, 3, and 4 features were analyzed for the test data sets.

FPC and classification error were computed for each subset.

Experiments (cont.)

FPC as a feature subset selection criterion

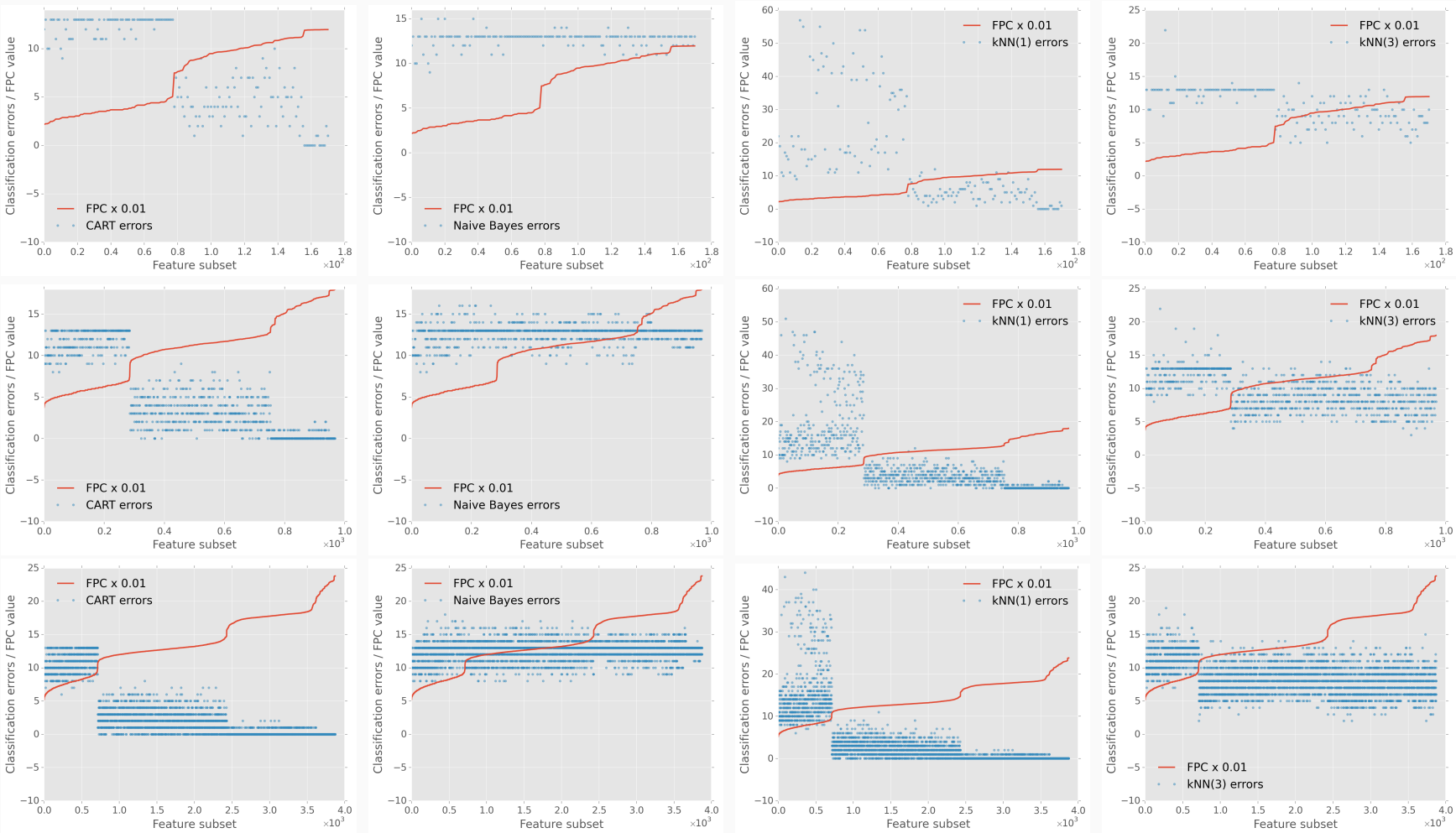
All possible subsets of 2, 3, and 4 features were analyzed for the test data sets.

FPC and classification error were computed for each subset.

Rank correlation (Spearman's ρ and $p-$value) between FPC and the number of classification errors ($p-$value shown in parentheses).

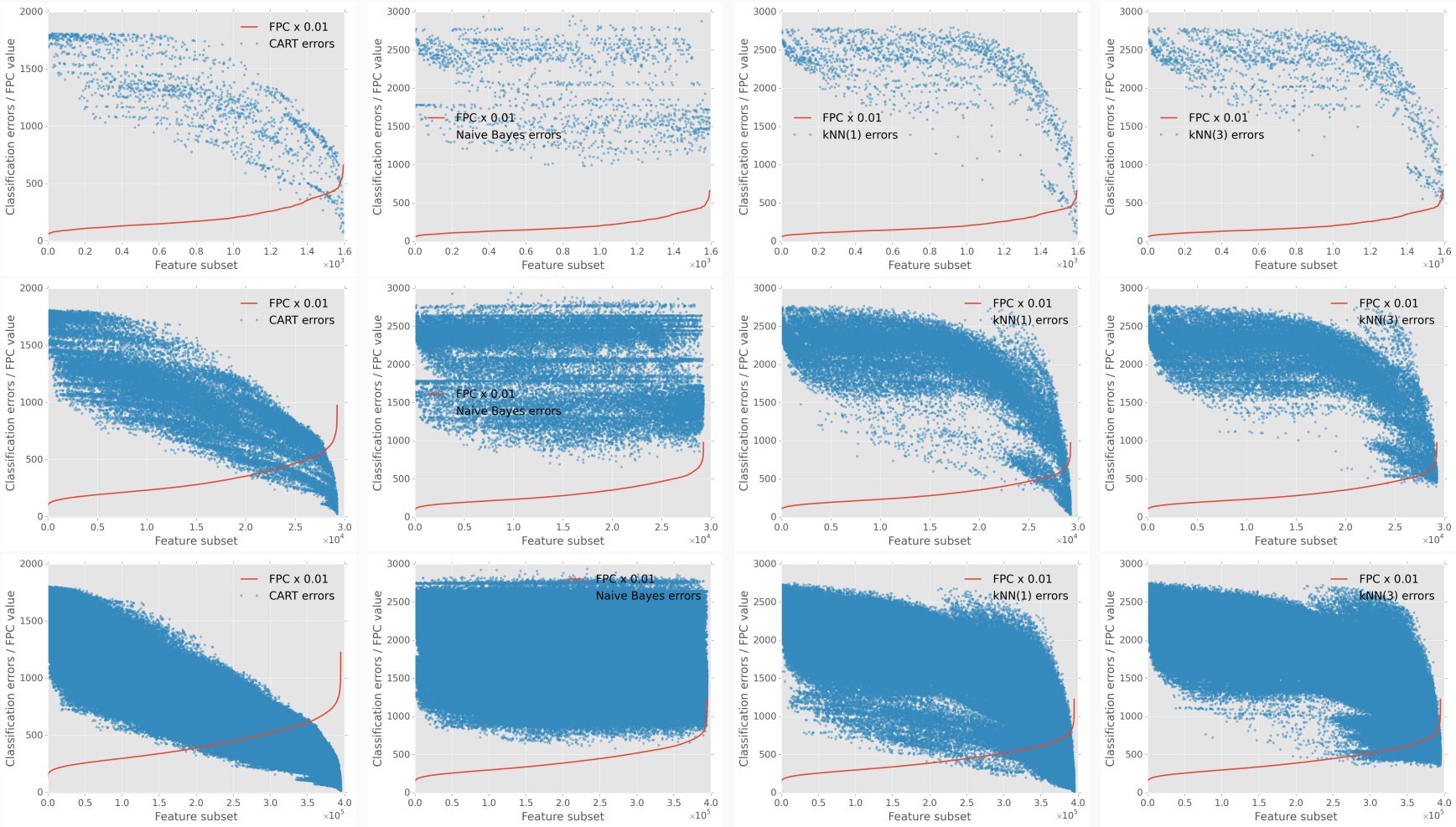
	Classifier			
Dataset	CART	NB	kNN(1)	kNN(3)
Hepatitis	-0.786 ($3.9\mathrm{E}^{-7}$)	-0.039 (0.6)	-0.781 ($2.2\mathrm{E}^{-36}$)	-0.688 ($2.5\mathrm{E}^{-25}$)
Spambase	-0.840 (0.0)	-0.300 ($1.2\mathrm{E}^{-34}$)	-0.534 ($1.7\mathrm{E}^{-118}$)	-0.530 ($4.6\mathrm{E}^{-116}$)

Experiments (cont.)



FPC vs. classification errors in the Hepatitis data set

Experiments (cont.)



FPC vs. classification errors in the Spambase data set

Experiments (cont.)

GA and DE as feature subset selection metaheuristics

Both methods executed with the best parameters found by trial-and-error runs, a total of 10,000 ff. evaluations each, 50 independent runs.

Experiments (cont.)

GA and DE as feature subset selection metaheuristics

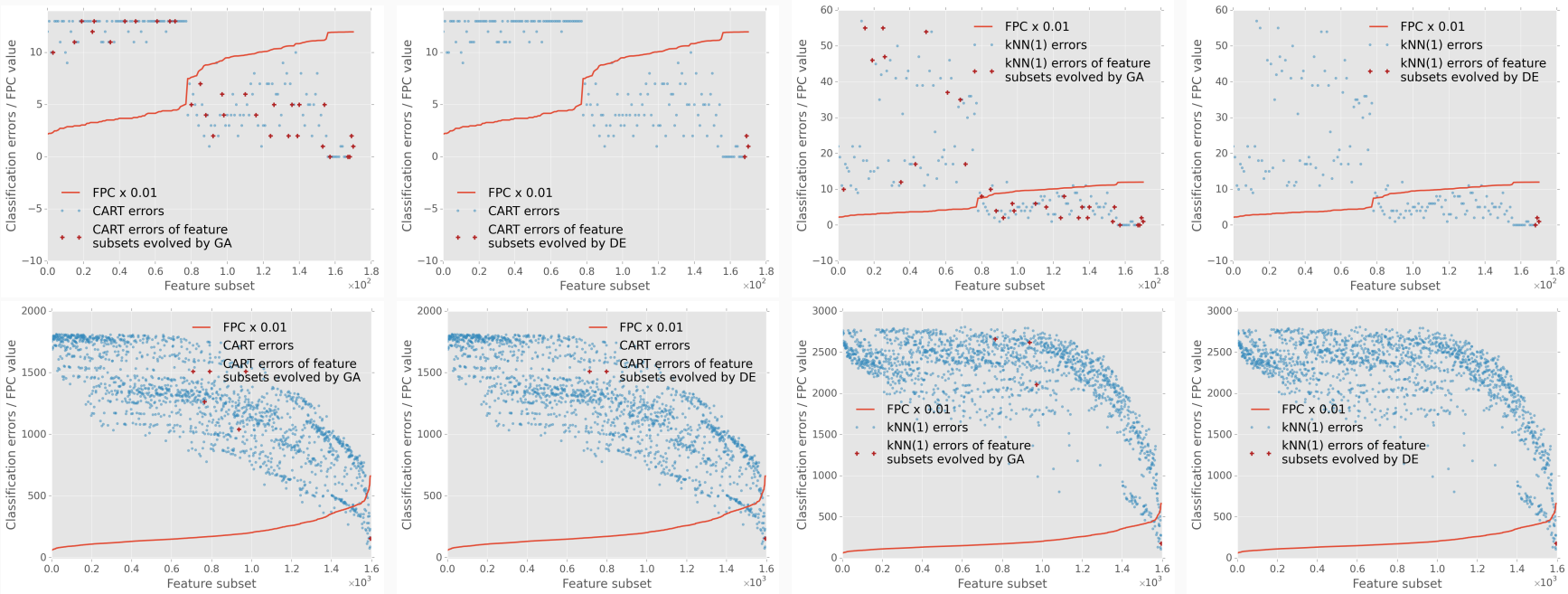
Both methods executed with the best parameters found by trial-and-error runs, a total of 10,000 ff. evaluations each, 50 independent runs.

The percent of feature subsets with FPC lower than best, average, and worst subsets found by the investigated methods.

		GA percentile			DE percentile		
Dataset	k	best	average	worst	best	average	worst
Hepatitis	2	99.42	57.89	2.34	99.42	99.42	99.42
	3	100.00	94.22	24.10	100.00	100.00	100.00
	4	99.96	97.81	33.13	99.96	99.96	99.96
Spambase	2	100.00	99.81	47.99	100.00	100.00	100.00
	3	100.00	99.99	4.97	100.00	100.00	100.00
	4	100.00	100.00	100.00	100.00	99.99	99.99

Note: all the best solutions have found feature subsets with maximum possible FPC

Experiments (cont.)



CART and kNN(1) classification errors of 2-feature subsets evolved by GA and DE on the *Hepatitis* (1st row) and *Spambase* data sets (2nd row).

Experiments (cont.)

The final FPC of feature subsets evolved by the GA and the DE.

		FPC of GA-evolved feature subsets			FPC of DE-evolved feature subsets		
Dataset	k	best	average (σ)	worst	best	average (σ)	worst
Hepatitis	2	1195	939.52 (331.43)	230	1195	1195 (0)	1195
	3	1796	1694.94 (255.15)	646	1796	1796 (0)	1796
	4	2380	2274.38 (294.86)	1238	2380	2380 (0)	2380
	5	2972	2887.30 (303.79)	1317	2972	2972 (0)	2972
	10	4728	4677.40 (277.75)	2743	4728	4727.90 (0.30)	4727
	15	5544	5261.40 (457.61)	3989	5544	5518.04 (32.31)	5452
Spambase	2	66064	63203.02 (11328.08)	16671	66064	66064 (0)	66064
	3	97466	95822.56 (11504.08)	15294	97466	97466 (0)	97466
	4	122431	122431 (0)	122431	122431	122318.92 (549.08)	119629
	5	142234	142234 (0)	142234	142234	142110.56 (604.73)	139148
	10	228155	221059.80 (5283.04)	210622	217278	206335.58 (4840.57)	198413
	15	287258	276567.86 (7387.49)	258259	274438	260328.52 (5225.25)	251003

Experiments (cont.)

The final FPC of feature subsets evolved by the GA and the DE.

		FPC of GA-evolved feature subsets			FPC of DE-evolved feature subsets		
Dataset	k	best	average (σ)	worst	best	average (σ)	worst
Hepatitis	2	1195	939.52 (331.43)	230	1195	1195 (0)	1195
	3	1796	1694.94 (255.15)	646	1796	1796 (0)	1796
	4	2380	2274.38 (294.86)	1238	2380	2380 (0)	2380
	5	2972	2887.30 (303.79)	1317	2972	2972 (0)	2972
	10	4728	4677.40 (277.75)	2743	4728	4727.90 (0.30)	4727
	15	5544	5261.40 (457.61)	3989	5544	5518.04 (32.31)	5452
Spambase	2	66064	63203.02 (11328.08)	16671	66064	66064 (0)	66064
	3	97466	95822.56 (11504.08)	15294	97466	97466 (0)	97466
	4	122431	122431 (0)	122431	122431	122318.92 (549.08)	119629
	5	142234	142234 (0)	142234	142234	142110.56 (604.73)	139148
	10	228155	221059.80 (5283.04)	210622	217278	206335.58 (4840.57)	198413
	15	287258	276567.86 (7387.49)	258259	274438	260328.52 (5225.25)	251003

Lesson learned

An efficient feature subset evaluation criterion based on a fast approximation of feature subset entropy was proposed and evaluated.

Results suggest that the fitness function based on FPC is reasonable – feature subsets with high values of FPC correspond to feature subsets that yield low classification error of test classifiers.

The DE performs better for small data and/or low–dimensional feature subsets while the GA seems to be more suitable for large data and larger feature subsets.

Deep feature selection

Deep learning

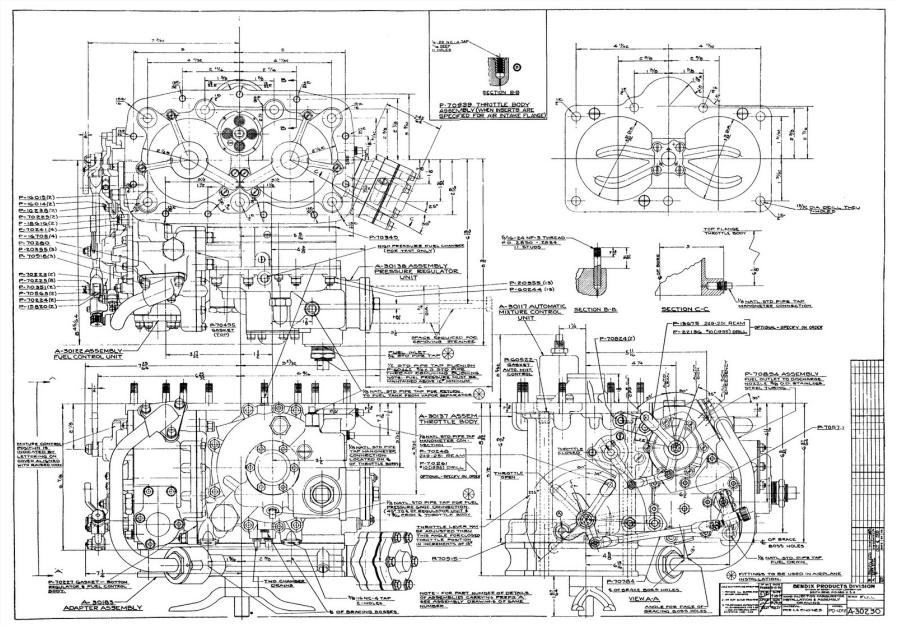
Deep learning solves the representation learning problem by introducing representations that are expressed in terms of other, (simpler) representations.

Creates a hierarchy of representations. More complex concepts are defined as composition of simpler ones and a clear interpretation of filters is desired.

Deep learning

Deep learning solves the representation learning problem by introducing representations that are expressed in terms of other, (simpler) representations.

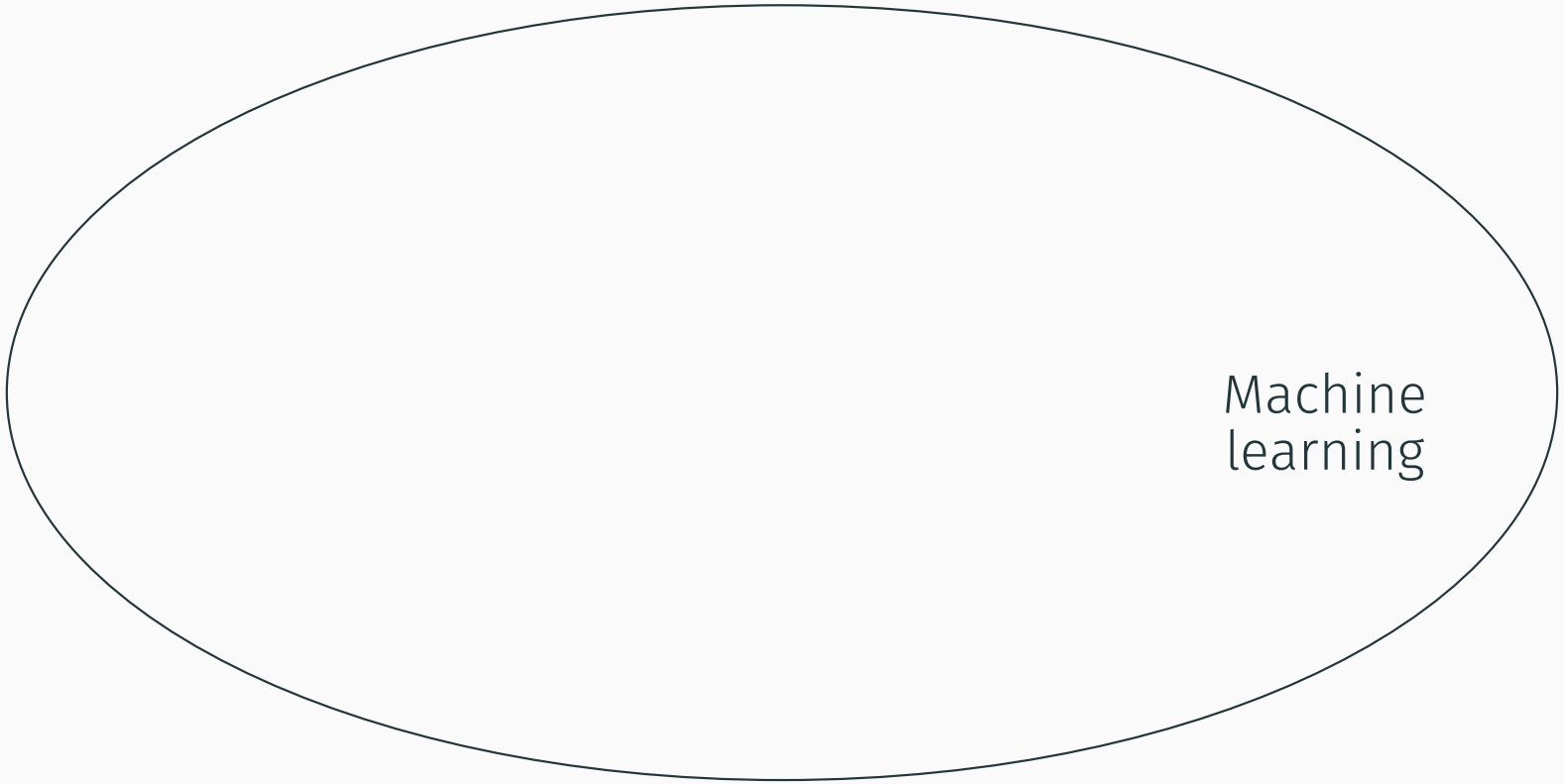
Creates a hierarchy of representations. More complex concepts are defined as composition of simpler ones and a clear interpretation of filters is desired.



Deep learning

Deep learning is a high–level approach that solves the representation learning problem by introducing representations that are expressed in terms of other, (simpler) representations.

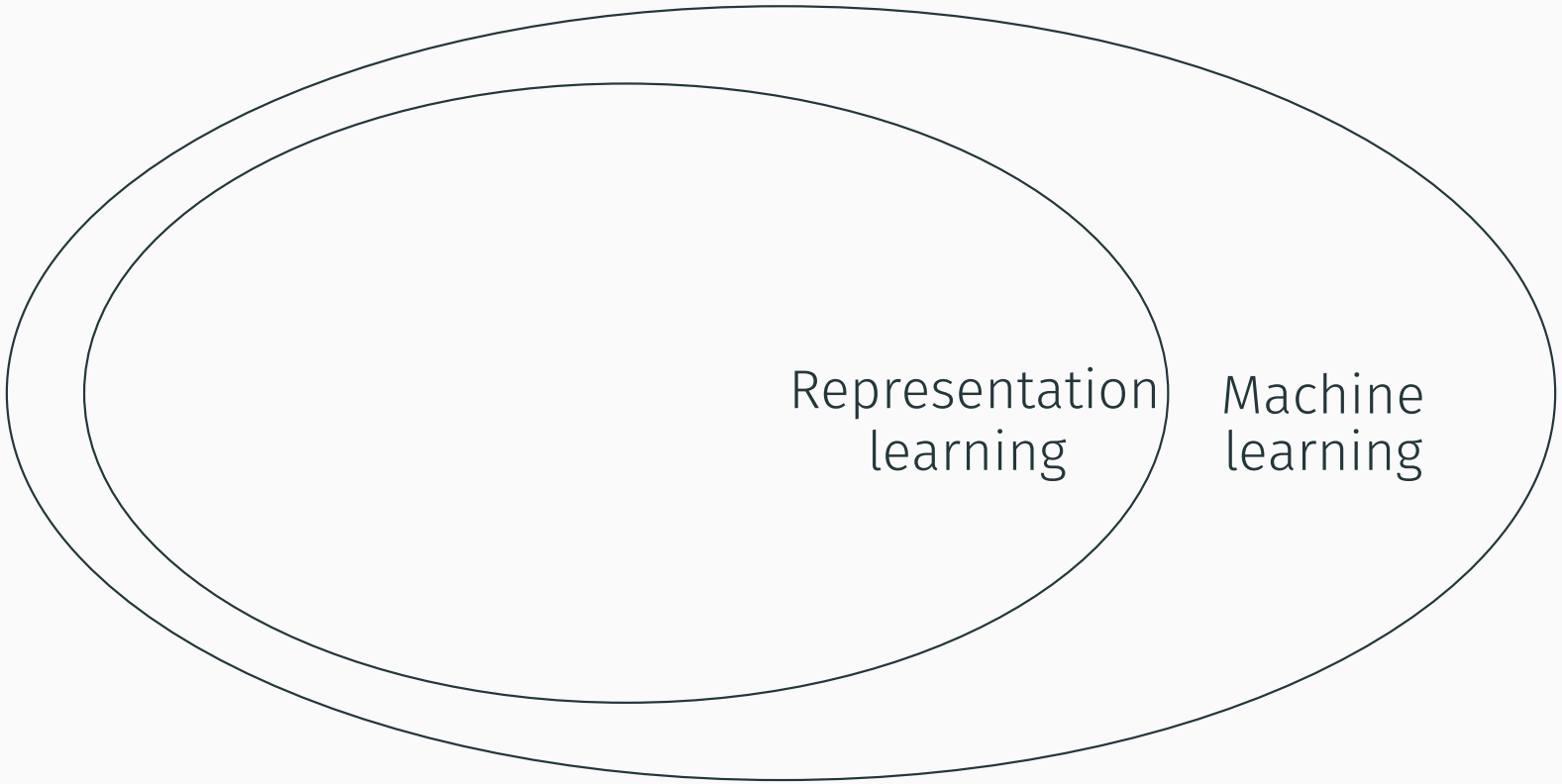
Creates a hierarchy of representations. More complex concepts are defined as composition of simpler ones and a clear interpretation of filters is desired.



Deep learning

Deep learning is a high–level approach that solves the representation learning problem by introducing representations that are expressed in terms of other, (simpler) representations.

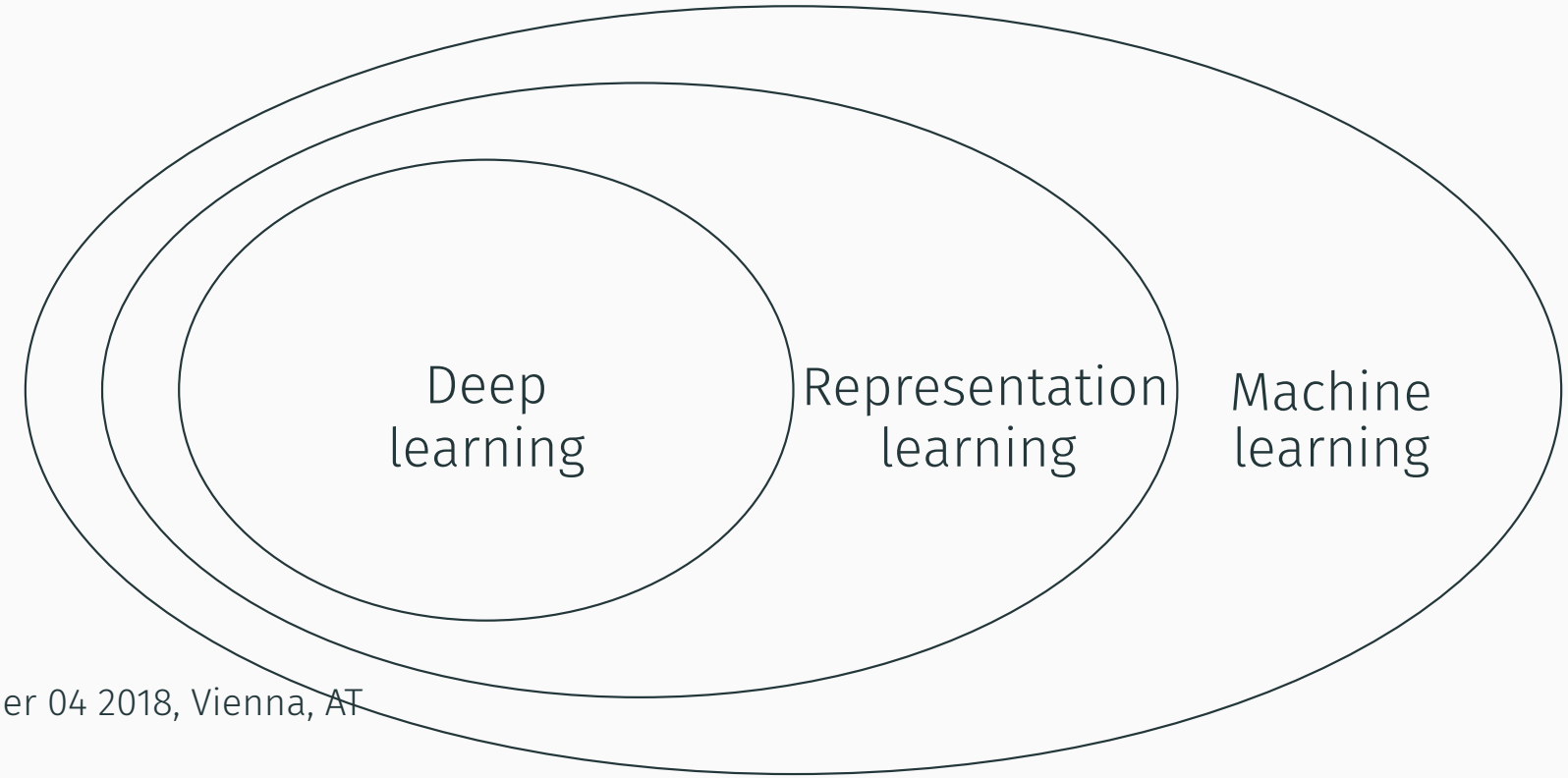
Creates a hierarchy of representations. More complex concepts are defined as composition of simpler ones and a clear interpretation of filters is desired.



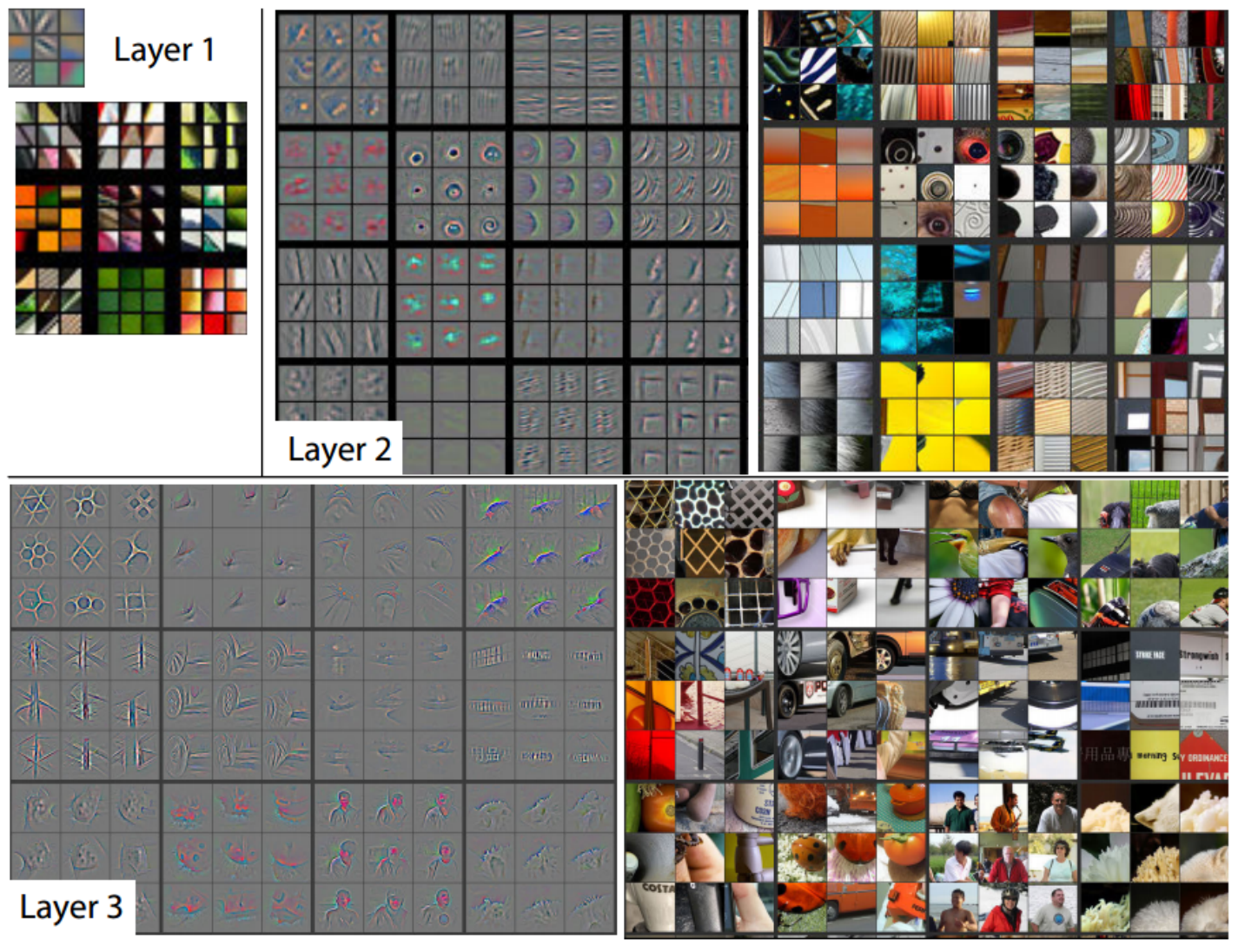
Deep learning

Deep learning is a high–level approach that solves the representation learning problem by introducing representations that are expressed in terms of other, (simpler) representations.

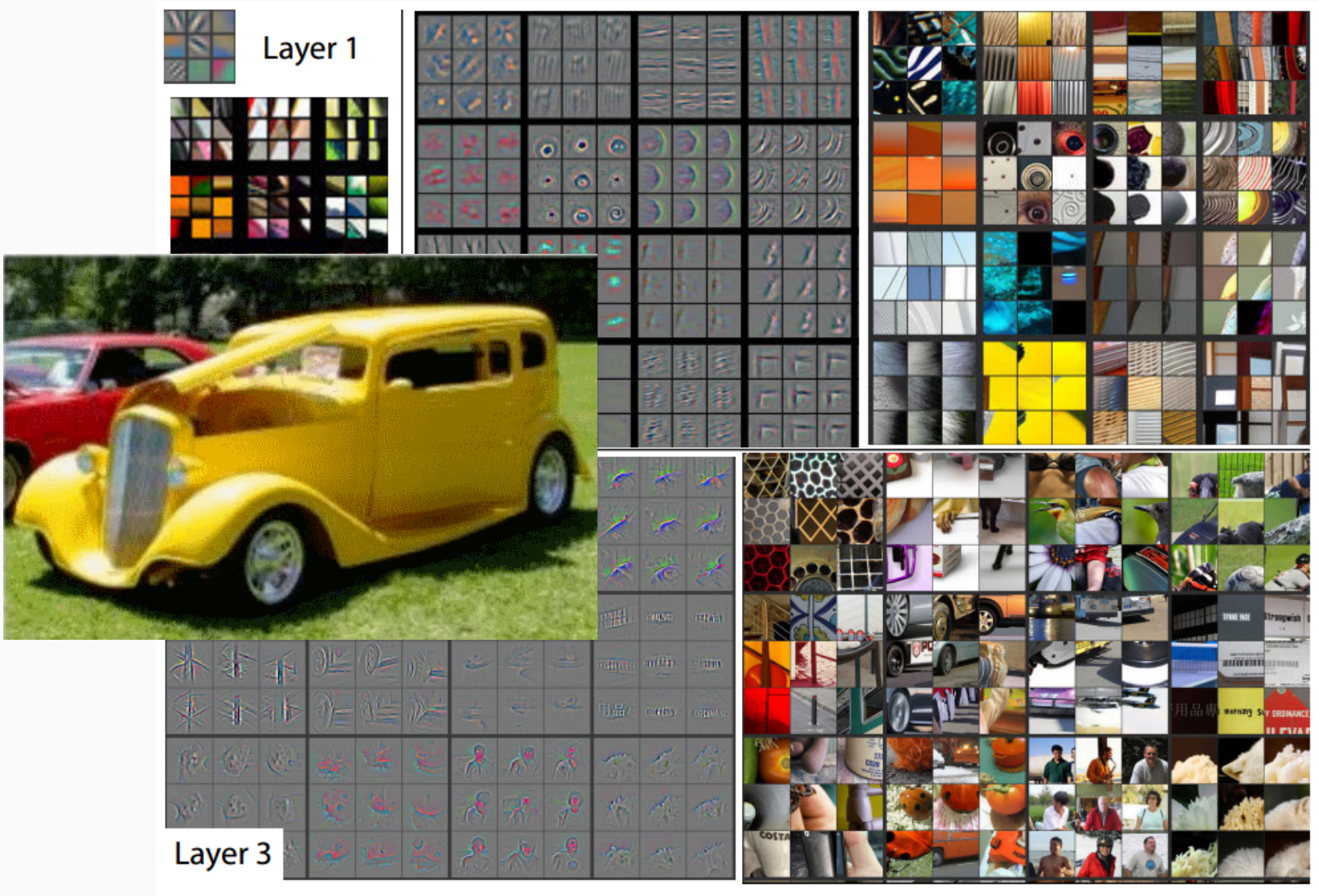
Creates a hierarchy of representations. More complex concepts are defined as composition of simpler ones and a clear interpretation of filters is desired.



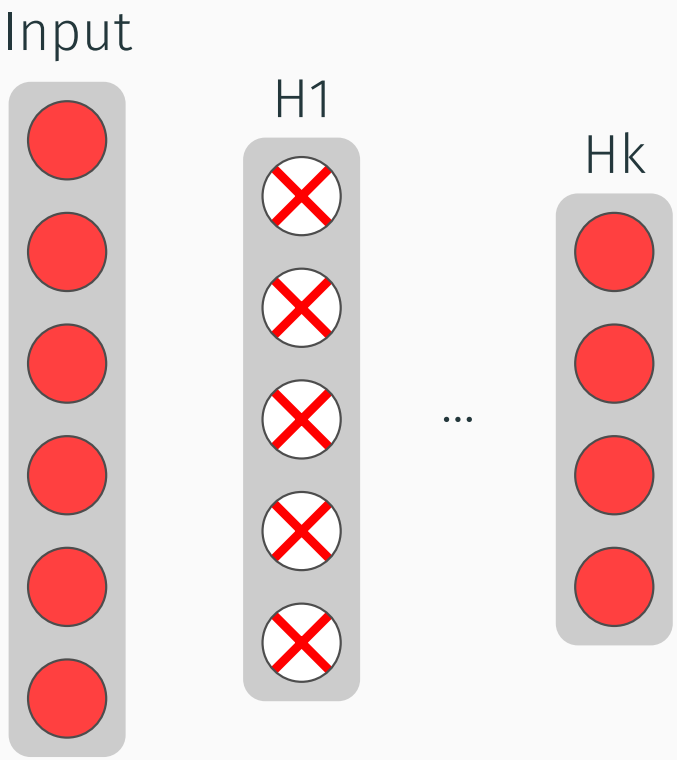
Example: Convolutional neural network (visualization)



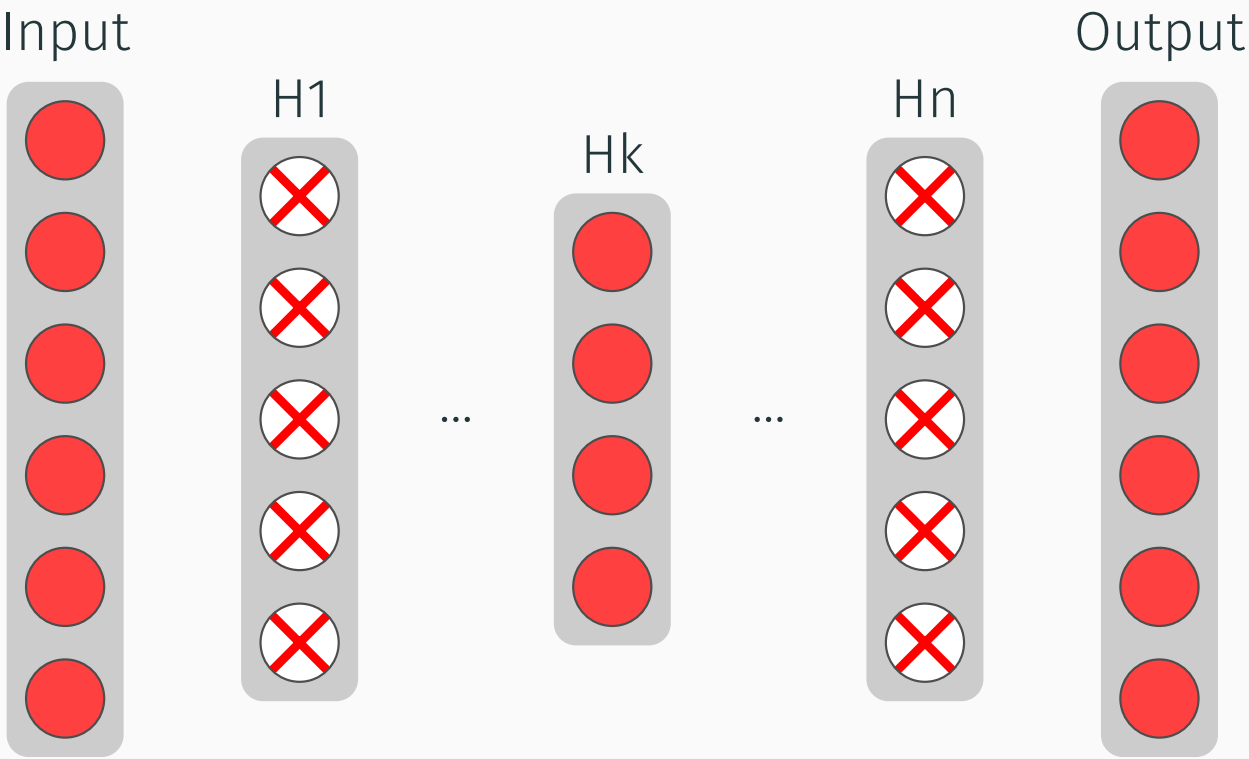
Example: Convolutional neural network (visualization)



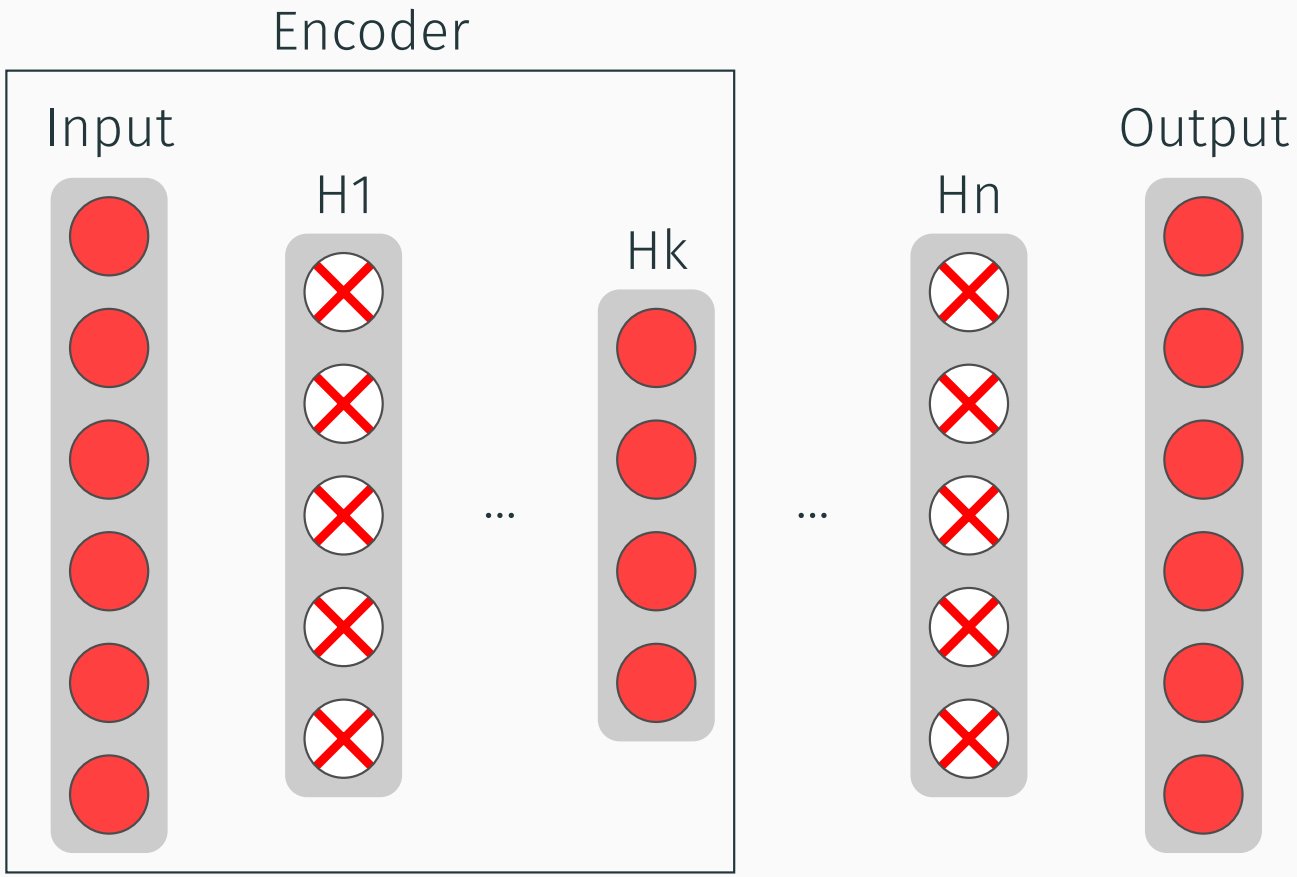
Example: Representation learning (autoencoder/Diablo network)



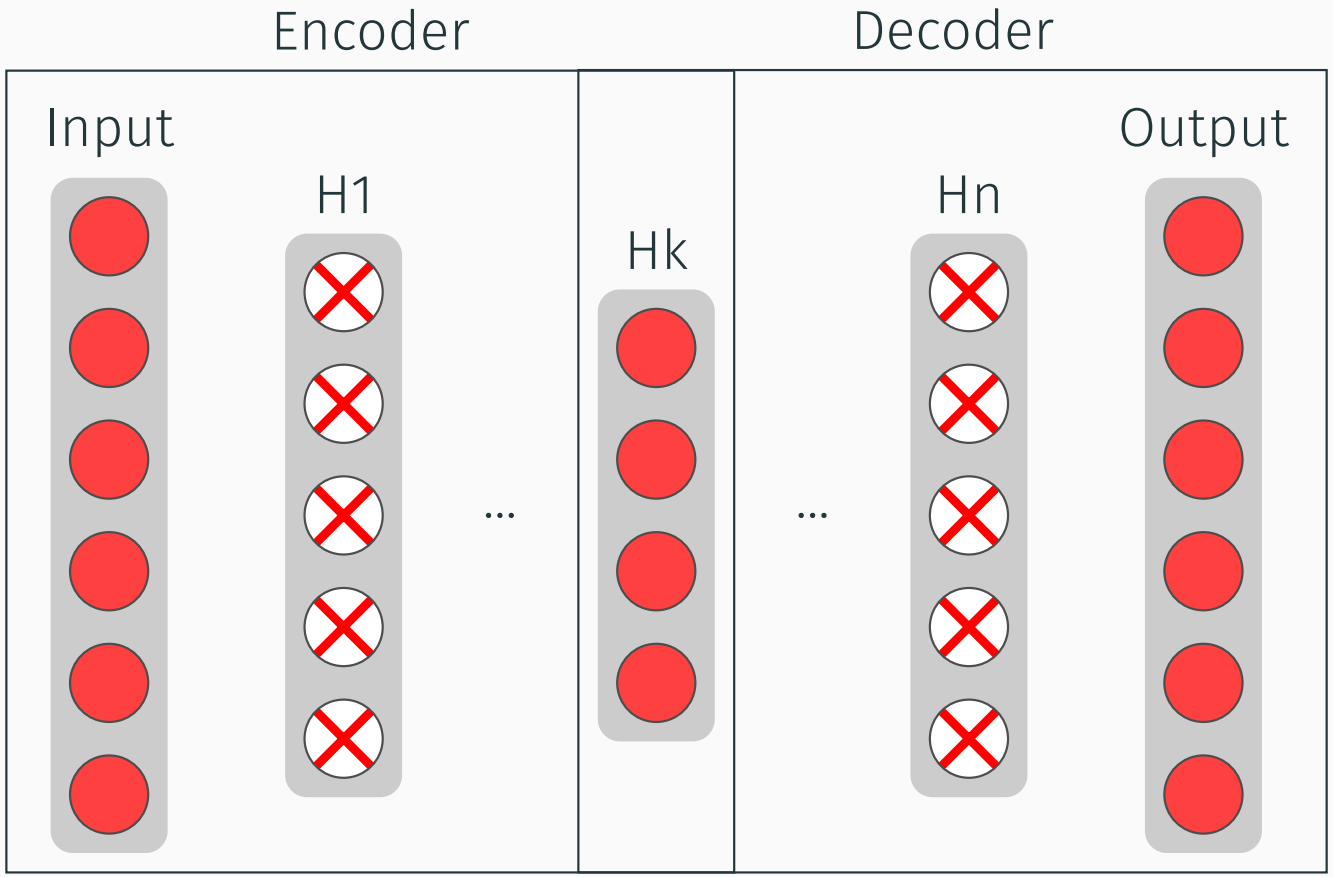
Example: Representation learning (autoencoder/Diablo network)



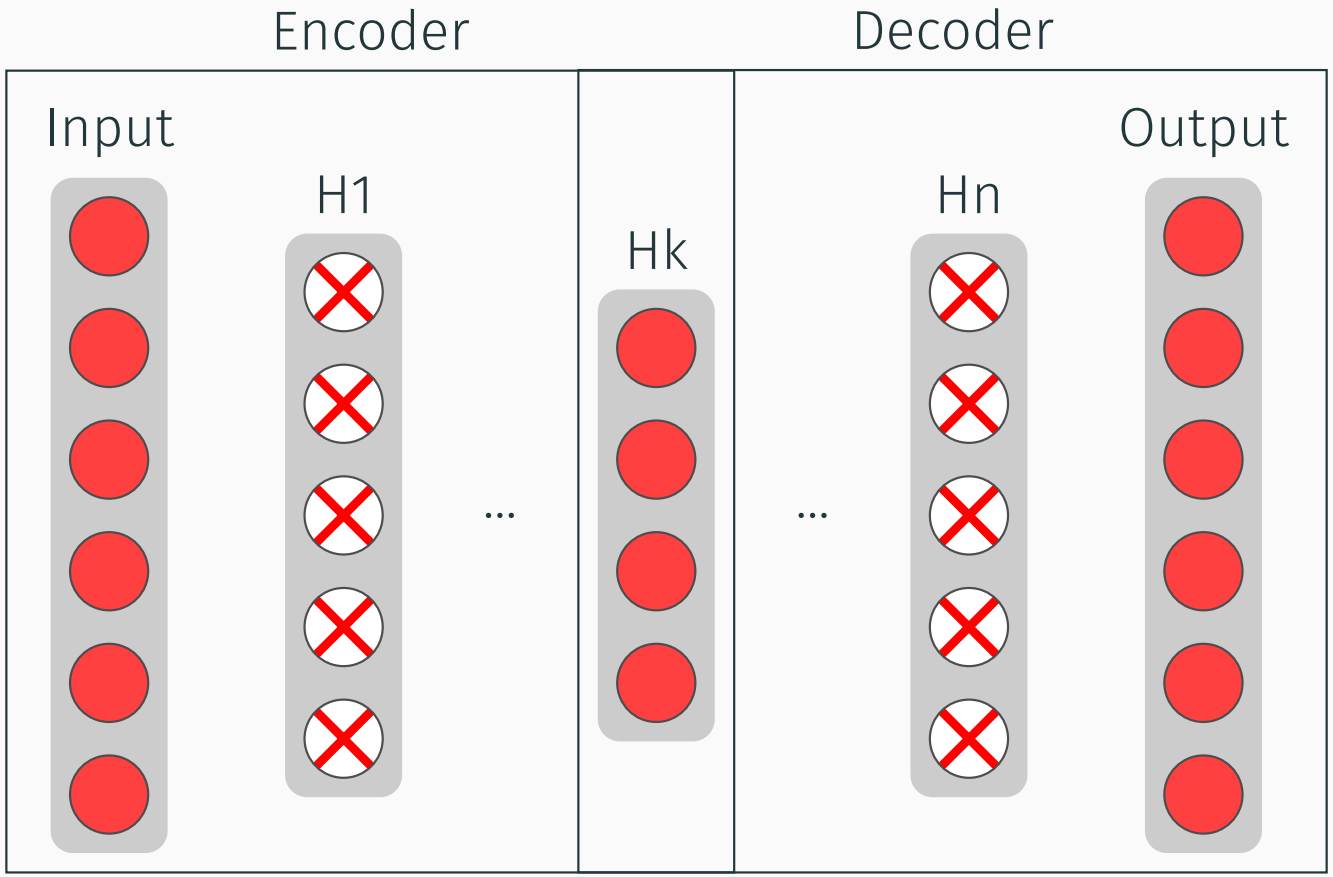
Example: Representation learning (autoencoder/Diablo network)



Example: Representation learning (autoencoder/Diablo network)



Example: Representation learning (autoencoder/Diablo network)



Code = Representation

Deep feature selection

Deep feature selection (DFS) is a family of methods that use the principles of deep learning for feature selection.

Deep feature selection

Deep feature selection (DFS) is a family of methods that use the principles of deep learning for feature selection.

They seek a higher–level representation of features (HLF) and try to utilize them, directly or indirectly, in the feature selection process.

Deep feature selection

Deep feature selection (DFS) is a family of methods that use the principles of deep learning for feature selection.

They seek a higher–level representation of features (HLF) and try to utilize them, directly or indirectly, in the feature selection process.

Example

Sentiment is a higher–level feature of texts (e.g. reviews). It can be learned in a semi–supervised manner via an algoritihm (Active Deep Network) based on Restricted Bolzmann Machines (Ruangkanokmas et al., 2016).

Deep feature selection

Deep feature selection (DFS) is a family of methods that use the principles of deep learning for feature selection.

They seek a higher–level representation of features (HLF) and try to utilize them, directly or indirectly, in the feature selection process.

Example

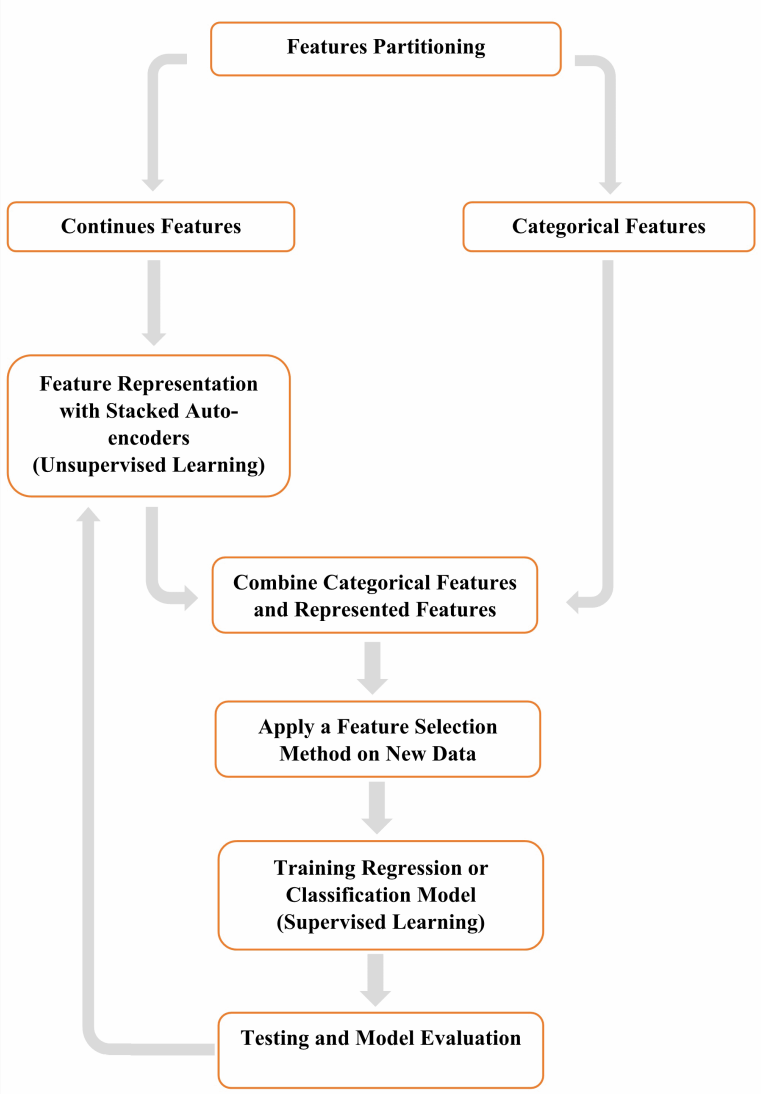
Sentiment is a higher–level feature of texts (e.g. reviews). It can be learned in a semi–supervised manner via an algoritihm (Active Deep Network) based on Restricted Bolzmann Machines (Ruangkanokmas et al., 2016).

DFS with HLF $\times$ DFS as a reconstruction problem $\times$ DFS via weight learning

Feature selection with higher–level features

HLF as an input for feature selection

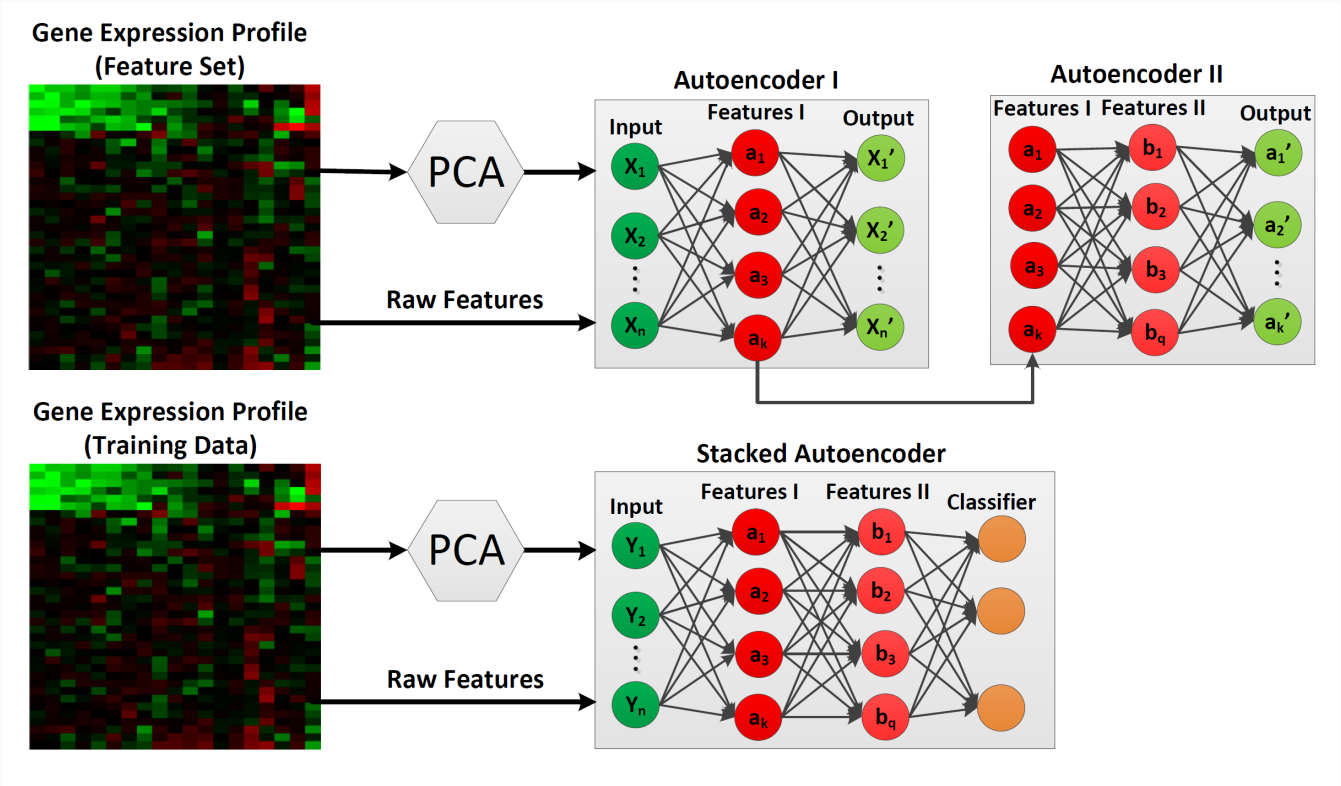
Higher–level features can be used to replace continous features for a standard feature selection algorithm (Nezhad et al. 2016).



Feature selection with higher–level features (cont.)

HLF in a data transformation pipeline

Data dimension is first reduced by PCA. Deep sparse encoding of the data is obtained (via stacked autoencoders). In the end, the learned higher–level features are used together with original features (raw data) for classification (Fakoor et al., 2013)



Deep feature selection as a reconstruction problem

An autoencoder/deep belief network is used to learn a sparse representation (code) of input features.

Deep feature selection as a reconstruction problem

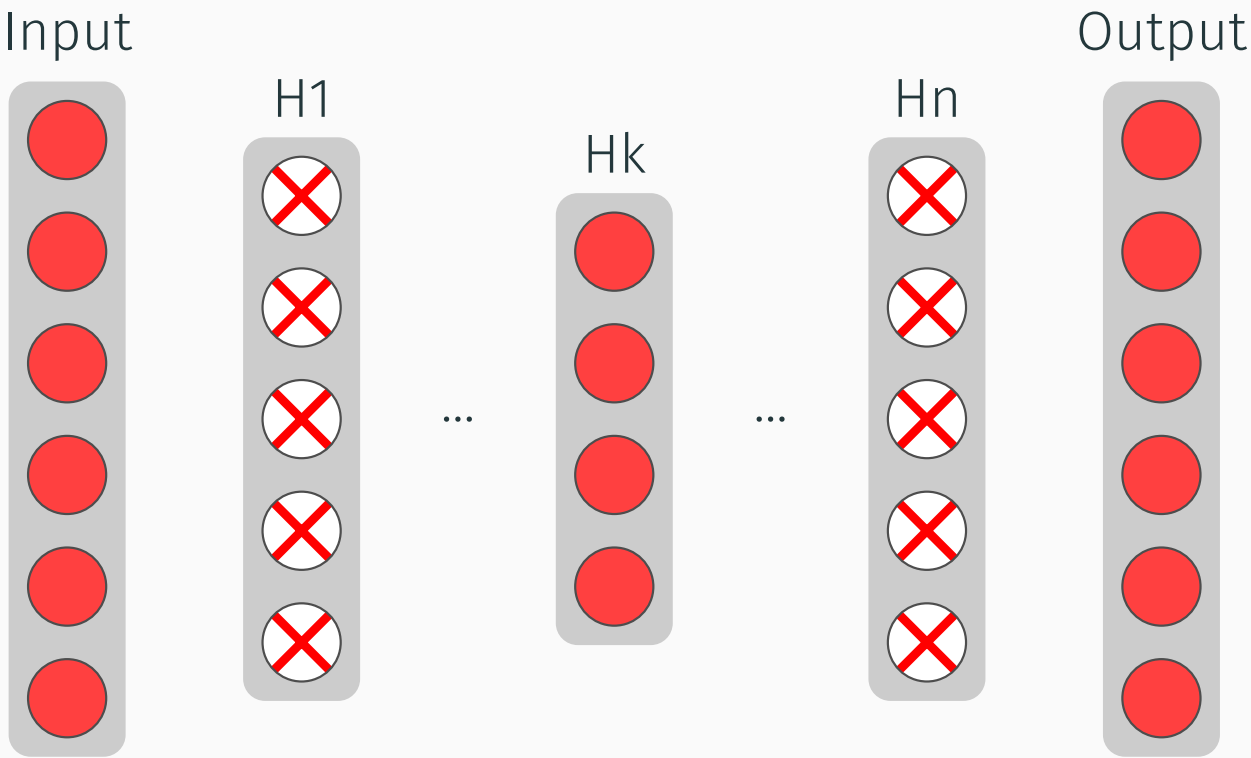
An autoencoder/deep belief network is used to learn a sparse representation (code) of input features.

Features with low reconstruction error are selected.

Deep feature selection as a reconstruction problem

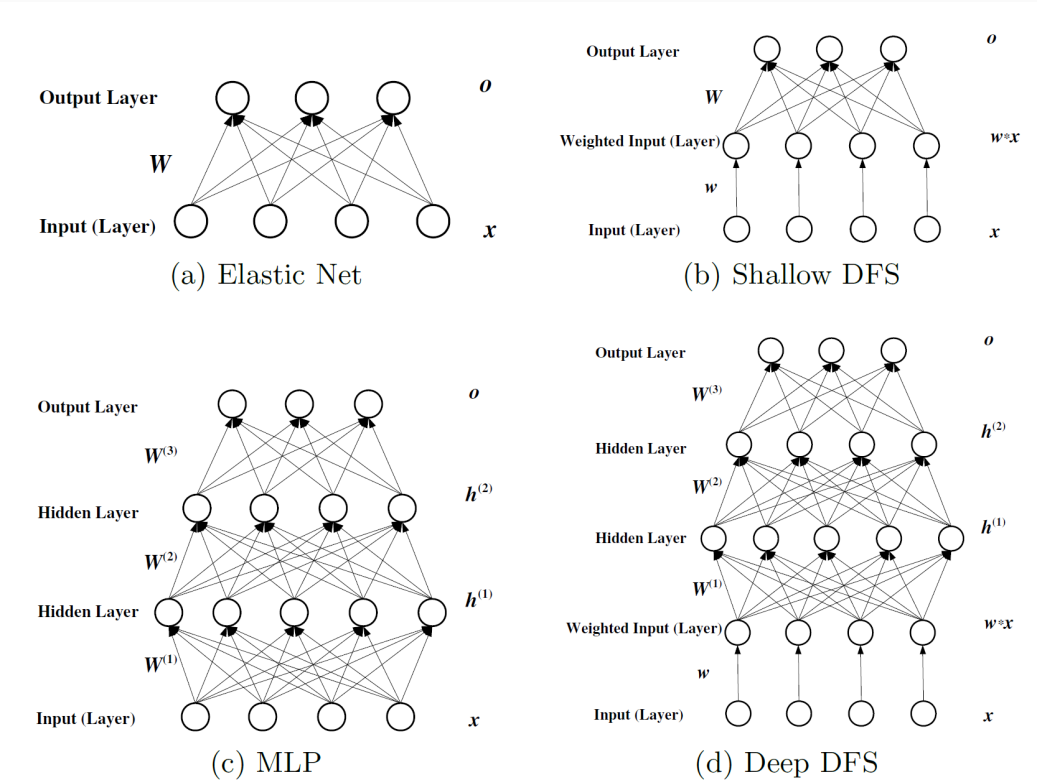
An autoencoder/deep belief network is used to learn a sparse representation (code) of input features.

Features with low reconstruction error are selected.



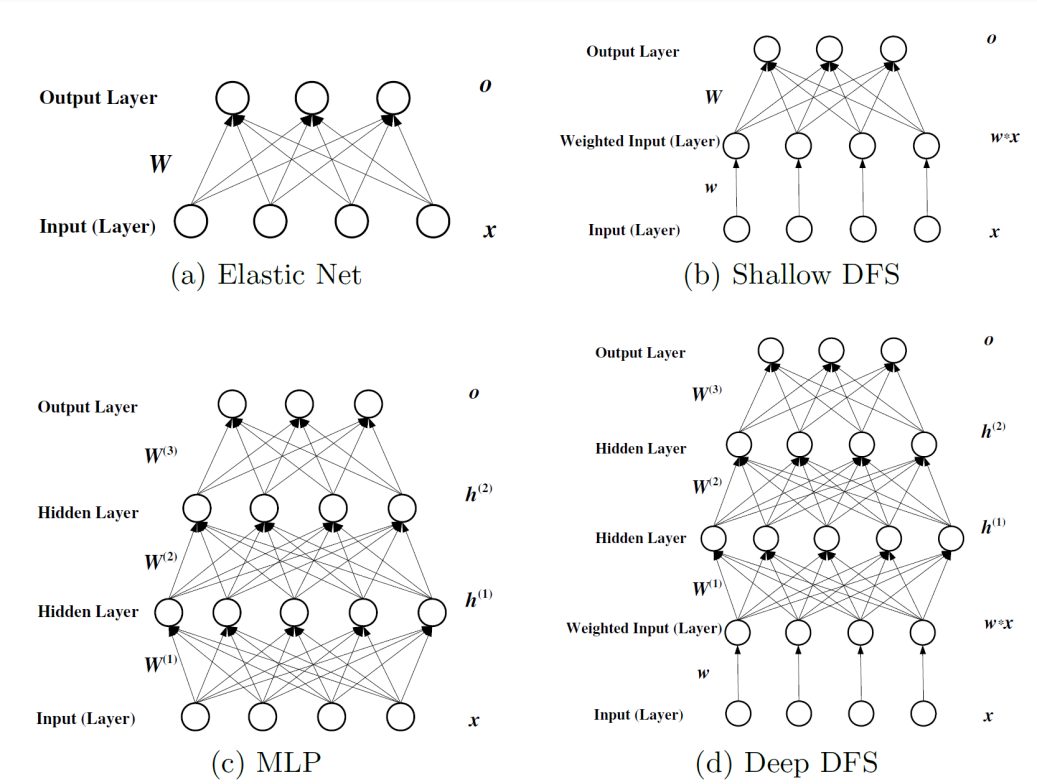
Feature selection via weight learning

A neural model can be augmented by an additional layer that serves as a sparse one-to-one linear connection between input and first hidden layer (Li et al., 2016).



Feature selection via weight learning

A neural model can be augmented by an additional layer that serves as a sparse one-to-one linear connection between input and first hidden layer (Li et al., 2016).



Most important features are those with high weights after training.

Summary

Summary

Feature selection is an important data (pre)processing step.

Summary

Feature selection is an important data (pre)processing step.

A variety of nature–inspired methods can be used to implement efficient feature selection schemes.

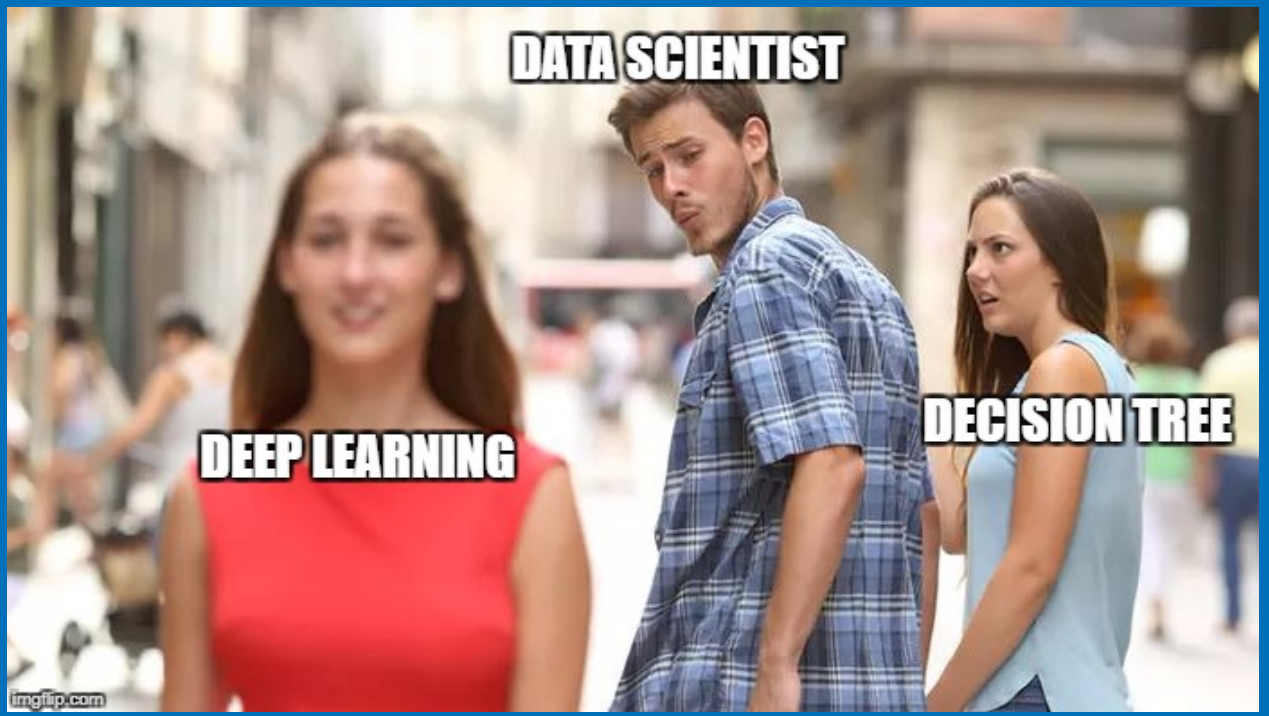
Summary

Feature selection is an important data (pre)processing step.

A variety of nature–inspired methods can be used to implement efficient feature selection schemes.

Deep feature selection is one of the hot topics in this area, bringing new opportunities and research challenges.

Data scientist’s life is full of wonderful options!



References

1. Pavel Krömer, Jan Platos, Jana Nowaková, Václav Snásel: Optimal column subset selection for image classification by genetic algorithms. Annals OR 265(2): 205-222 (2018)
2. Pavel Krömer, Jan Platos: Genetic algorithm for entropy-based feature subset selection. CEC 2016: 4486-4493
3. Pavel Krömer, Jan Platos: Evolutionary Feature Subset Selection with Compression-based Entropy Estimation. GECCO 2016: 933-940
4. Y. Li, C.Y. Chen, and W. Wasserman, Deep Feature Selection: Theory And Application To Identify Enhancers And Promoters, Journal of Computational Biology , vol. 23, pp. 322-336, 2016.
5. P. Ruangkanokmas, T. Achalakul, and K. Akkarajitsakul, Deep Belief Networks With Feature Selection For Sentiment Classification, 7th International Conference on Intelligent Systems, Modeling and Simulation, 2016.
6. R. Fakoor, f. Ladhak, A. Nazi, and M. Huber, Using Deep Learning To Enhance Cancer Diagnosis And Classification, 30th International Conference on Machine Learning, 2013.