Deep learning

Machine learning overview

Hamid Beigy

Sharif University of Technology

September 27, 2024

Table of contents

What is machine learning?

What is machine learning?

The field of *machine learning* is concerned with the question of how to construct computer programs that automatically improve with the experience.

Definition (Mohri, Rostamizadeh, and Talwalkar 2018)

Computational methods that use experience to improve performance or to make accurate predictions.

Definition (Mitchell 1997)

A computer program is said to *learn*

1. from *training experience* E
2. with respect to some class of *tasks* T
3. and *performance measure* P,

if its performance at tasks in T, measured by P, improves with experience E.

What is machine learning? (Example)

Example (Checkers learning problem)

Class of task T: playing checkers.

Performance measure P: percent of games won against opponents.

Training experience E: playing practice game against itself.

Example (Handwriting recognition learning problem)

Class of task T: recognizing and classifying handwritten words within images.

Performance measure P: percent of words classified correctly.

Training experience E: a database of handwritten words with given classifications.

Why we need machine learning?

We need machine learning because

1. Tasks are too complex to program
   1.1 Tasks performed by animals/humans such as driving, speech recognition, image understanding, and etc.
   1.2 Tasks beyond human capabilities such as weather prediction, analysis of genomic data, web search engines, and etc.
2. Some tasks need adaptivity. When a program has been written down, it stays unchanged. In some tasks such as optical character recognition and speech recognition, we need the behavior to be adapted when new data arrives.

Types of machine learning

Machine learning algorithms based on the information provided to the learner can be classified into four main groups.

1. Supervised/predictive learning: The goal is to learn a mapping from inputs x to outputs y given the labeled set $S = \{(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)\}$.
2. Unsupervised/descriptive learning: The goal is to find interesting pattern in data $S = \{x_1, x_2, \ldots, x_N\}$. Unsupervised learning is arguably more typical of human and animal learning.
3. Representation Learning: The goal is to learn a good representation for every input raw pattern.
4. Reinforcement learning: Reinforcement learning is learning by interacting with an environment. A reinforcement learning agent learns from the consequences of its actions.

Applications of machine learning

1. Supervised learning:
   1.1 Classification:
      - Document classification and spam filtering.
      - Image classification and handwritten recognition.
      - Face detection and recognition.
   1.2 Regression:
      - Predict stock market price/ temperature of a location/ the amount of PSA.
2. Unsupervised/descriptive learning:
   - Discovering clusters/latent factors.
   - Discovering graph structures (correlation of variables).
   - Matrix completion (filling missing values).
   - Market-basket analysis (frequent item-set mining).
3. Representation learning:
   - Representing text, image, voice signals, graph nodes, etc.
4. Reinforcement learning:
   - Game playing and robot navigation.

Supervised learning

Supervised learning

1. In supervised learning, the goal is to find a mapping from inputs X to outputs t given a labeled set of input-output pairs

$$S = \{(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)\}.$$

   S is called training set.

2. In the simplest setting, each training input x is a $D-$dimensional vector of numbers.
3. Each component of x is called feature and x is called feature vector.
4. In general, x could be a complex structure of object, such as an image, a sentence, an email message, a time series, a molecular shape, a graph.
5. When $t_i \in \{1, 2, \ldots, M\}$, the problem is known as classification.
6. When $t_i \in \mathbb{R}$, the problem is known as regression.

Classification

1. In classification, the goal is to find a mapping from inputs X to outputs t, where $t \in \{1, 2, \ldots, M\}$ with M being the number of classes.
2. When $M = 2$, the problem is called binary classification. In this case, we often assume that $t \in \{-1, +1\}$ or $t \in \{0, 1\}$.
3. When $M > 2$, the problem is called multi-class classification.
4. Each sample is labeled as

$$h(x) = \begin{cases} 1 & \text{if positive example} \\ 0 & \text{if negative example} \end{cases}$$

5. Each sample in the training set is represented by an ordered pair (x, t) and the training set containing

$$S = \{(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)\}.$$

Classification (Cont.)

1. The learning algorithm should find a particular hypotheses $h \in H$ to approximate concept $c \in C$ as closely as possible.
2. The expert defines the hypothesis class H, but he can not say h.
3. We choose H and the aim is to find $h \in H$ that is similar to c. This reduces the problem of learning the class to the easier problem of finding the parameters that define h.
4. Hypothesis h makes a prediction for an instance x in the following way.

$$h(x) = \begin{cases} 1 & \text{if } h \text{ classifies } x \text{ as an instance of a positive example} \\ 0 & \text{if } h \text{ classifies } x \text{ as an instance of a negative example} \end{cases}$$

Classification (Cont.)

1. In real life, we don't know $c(x)$ and hence cannot evaluate how well $h(x)$ matches $c(x)$.
2. We use a small subset of all possible values x as the training set as a representation of that concept.
3. Empirical error (risk)/training error is the proportion of training instances such that $h(x) \neq c(x)$.

$$E_S(h) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[h(x_i) \neq c(x_i)\right]$$

4. When $E_S(h) = 0$, h is called a consistent hypothesis with dataset S.
5. For any problem, we can find infinitely many h such that $E_S(h) = 0$. But which of them is better than for prediction of future examples?
6. This is the problem of generalization, that is, how well our hypothesis will correctly classify the future examples that are not part of the training set.

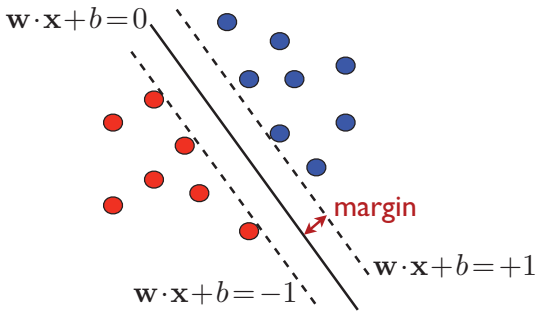
Classification (Generalization)

1. The generalization capability of a hypothesis usually measured by the true error/risk.

$$E(h) = \underset{x \sim D}{\textbf{Prob}}[h(x) \neq c(x)]$$

2. We assume that H includes C, that is there exists $h \in H$ such that $E_S(h) = 0$.
3. Given a hypothesis class H, it may be the cause that we cannot learn C; that is there is no $h \in H$ for which $E_S(h) = 0$.
4. Thus in any application, we need to make sure that H is flexible enough , or has enough capacity to learn C.

1. Support vector machine (SVM) is a linear classifier that maximizes the margine.



2. We need a classifier to maximize the margin subject to the constraints that all the training examples are classified correctly. Hence, we have

$$\textit{Minimize} \frac{1}{2}\|\mathbf{w}\|^2 \qquad \text{subject to} \qquad t_n\left(\langle \mathbf{w}, \mathbf{x}_n \rangle + b\right) \geq 1 \qquad \text{for all } n = 1, 2, \ldots, N$$

Logistic regression

1. Logistic regression is a model for probabilistic classification.
2. It predicts label probabilities rather than a hard value of the label.
3. The output of Logistic regression is a probability defined using the sigmoid function

$$P(C_1|x_n) = \sigma\left(\langle\theta, \mathbf{x}_n\rangle\right) = \frac{1}{1 + exp(-\langle\theta, \mathbf{x}_n\rangle)}$$

4. The loss function

$$E_S(t_n, h(x_n)) = \begin{cases} P(C_1|x_n) & t_n = 1 \\ 1 - P(C_1|x_n) & t_n = 0 \end{cases}$$

5. The loss function over training data is

$$E_S(\theta) = \sum_{n=1}^{N} [-t_n \ln P(C_1|x_n) - (1 - t_n) \ln(1 - P(C_1|x_n))]$$

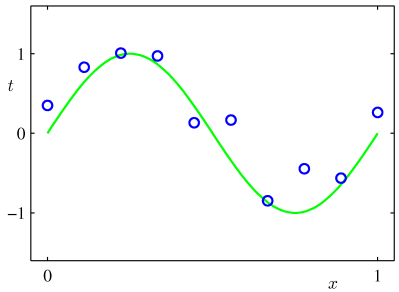
6. This loss function is called the cross-entropy loss.

1. $c(x)$ is a continuous function. Hence the training set is in the form of

$$S = \{(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)\}, t_k \in \mathbb{R}.$$

2. There is noise added to the output of the unknown function.

$$t_k = f(x_k) + \epsilon \qquad \forall k = 1, 2, \ldots, N$$



3. The explanation for the noise is that there are extra hidden variables that we cannot observe.

$$t_k = f^*(x_k, z_k) + \epsilon \qquad \forall k = 1, 2, \ldots, N$$

z_k denotes hidden variables

Regression(cont.)

1. Our goal is to approximate the output by function $g(x)$.
2. The empirical error on the training set S is

$$E_S(g) = \frac{1}{N} \sum_{k=1}^{N} [t_k - g(x_k)]^2$$

3. The aim is to find $g(.)$ that minimizes the empirical error.
4. We assume that a hypothesis class for $g(.)$ with a small set of parameters.

1. The training data is not sufficient to find the solution, we should make some extra assumption for learning.

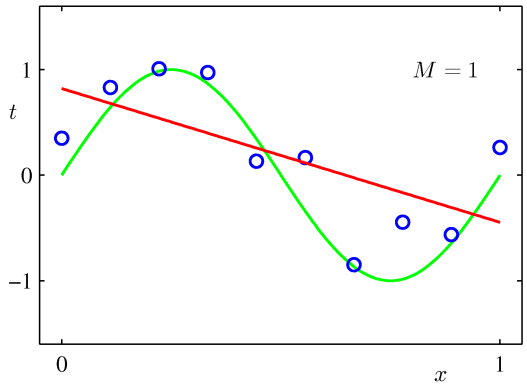
Inductive bias

The inductive bias of an algorithm is the set of assumptions that the learner uses to predict outputs given inputs that it has not encountered.

2. We assume a hypothesis class (inductive bias). Each hypotheses class has certain capacity and can learn only certain functions.
3. How to choose the right inductive bias (for example hypotheses class)? This is called model selection.
4. How well a model trained on the training set predicts the right output for new instances is called generalization.
5. For best generalization, we should choose the right model that matches the complexity of the hypothesis with the complexity of the function underlying data.

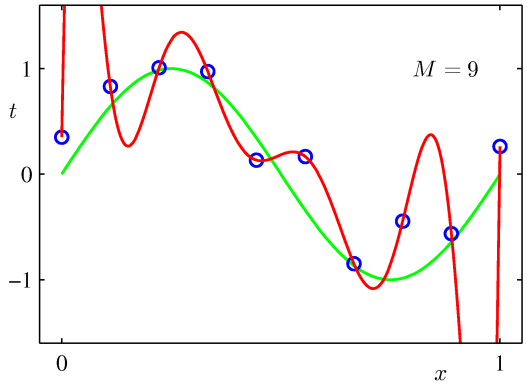
Model selection (cont.)

1. For best generalization, we should choose the right model that match the complexity of the hypothesis with the complexity of the function underlying data.
2. If the hypothesis is less complex than the function, we have underfitting



Model selection (cont.)

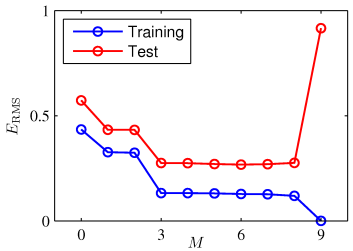
1. If the hypothesis is more complex than the function, we have overfitting



2. There are trade-off between three factors
   - 2.1 Complexity of hypotheses class
   - 2.2 Amount of training data
   - 2.3 Generalization error

Model selection (cont.)

1. As the amount of training data increases, the generalization error decreases.
2. As the capacity of the models increases, the generalization error decreases first and then increases.



3. We measure generalization ability of a model using a validation set.
4. The available data for training is divided to
   - 4.1 Training set
   - 4.2 Validation data
   - 4.3 Test data

Summary of supervised learning

1. The training set S
   - 1.1 A set of N i.i.d distributed data.
   - 1.2 The ordering of data is not important
   - 1.3 The instances are drawn from the same distribution $p(x, t)$.
2. In order to have successful learning, three decisions must take
   - 2.1 Select appropriate model ($g(x; \theta)$)
   - 2.2 Select appropriate loss function

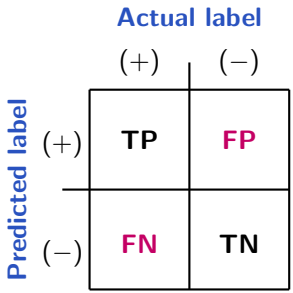
$$E_S(\theta) = \sum_k \ell(t_k, g(x_k; \theta))$$

   - 2.3 Select appropriate optimization procedure

$$\theta^* = \arg\min_\theta E_S(\theta)$$

Some performance measures of classifiers

1. What you can say about the accuracy of 90% or the error of 10% ?
2. For example, if $3-4\%$ of examples are from negative class, clearly accuracy of 90% is not acceptable.
3. Confusion matrix



4. Given M classes, a confusion matrix is a table of $M \times M$.

Some performance measures of classifiers

1. Precision (Positive predictive value) Precision is proportion of predicted positives which are actual positive and defined as

$$Precision(h) = \frac{TP}{TP + FP}$$

2. Recall (Sensitivity) Recall is proportion of actual positives which are predicted positive and defined as

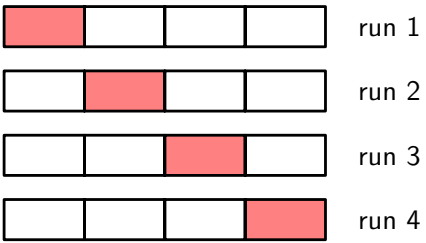
$$Recall(h) = \frac{TP}{TP + FN}$$

3. F_1-measure F_1-measure is harmonic mean between precision and recall and defined as

$$F_1(h) = 2 \times \frac{Persicion(h) \times Recall(h)}{Persicion(h) + Recall(h)}$$

Cross validation method

1. K-fold cross validation The initial data are randomly partitioned into K mutually exclusive subsets or folds, $S_1, S_2, \ldots, S_K$, each of approximately equal size. .



1.1 Training and testing is performed K times.

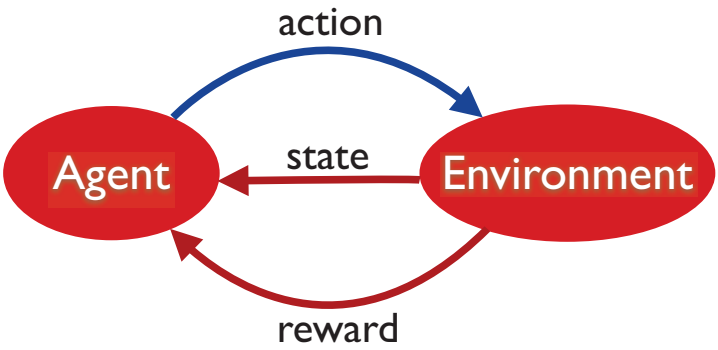
1.2 In iteration k, partition S_k is used for test and the remaining partitions collectively used for training.

1.3 The accuracy is the percentage of the total number of correctly classified test examples.

1.4 The advantage of K-fold cross validation is that all the examples in the dataset are eventually used for both training and testing.

Reinforcement learning

1. A key feature of reinforcement learning is that it explicitly considers the whole problem of a goal-directed agent interacting with an uncertain environment.



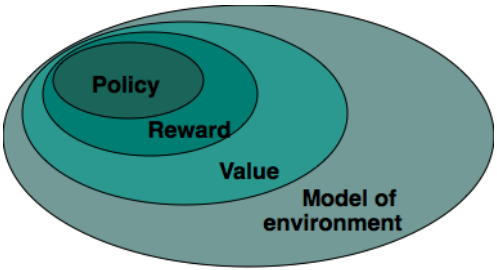
Elements of RL

Policy : A policy is a mapping from received states of the environment to actions to be taken (what to do?).

Reward function: It defines the goal of RL problem. It map each state-action pair to a single number called reinforcement signal, indicating the goodness of the action. (what is good?)

Value : It specifies what is good in the long run. (what is good because it predicts reward?)

Model of the environment (optional): This is something that mimics the behavior of the environment. (what follows what?)



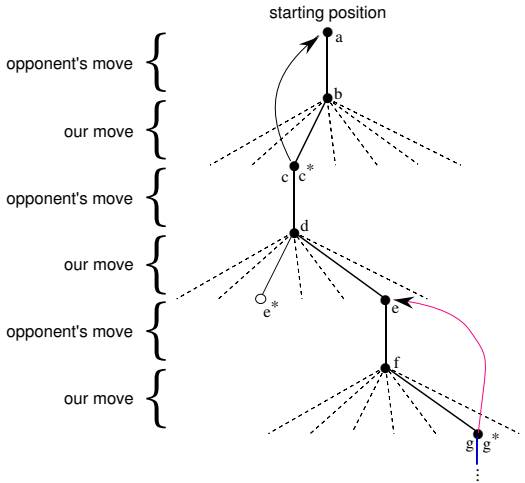
Reinforcement learning

1. Reinforcement learning is what how to map situations to actions so as to maximize a scalar reward/reinforcement signal
2. The learner is not told which actions to take as in supervised learning, but discover which actions yield the most reward by trying them.
3. The trial-and-error and delayed reward are the two most important feature of reinforcement learning.
4. Reinforcement learning is defined not by characterizing learning algorithms, but by characterizing a learning problem.
5. Any algorithm that is well suited for solving the given problem, we consider to be a reinforcement learning.
6. One of the challenges that arises in reinforcement learning and other kinds of learning is trade-off between exploration and exploitation.

An example : Tic-Tac-Toe

1. Consider a two-playes game (Tic-Tac-Toe)

X	O	O
O	X	X
		X



2. Consider the following updating

$$V(s) \leftarrow V(s) + \alpha[V(s') - V(s)]$$

Unsupervised learning

Unsupervised learning

1. Unsupervised learning is fundamentally problematic and subjective.
2. Examples :

   Clustering Find natural grouping in data.

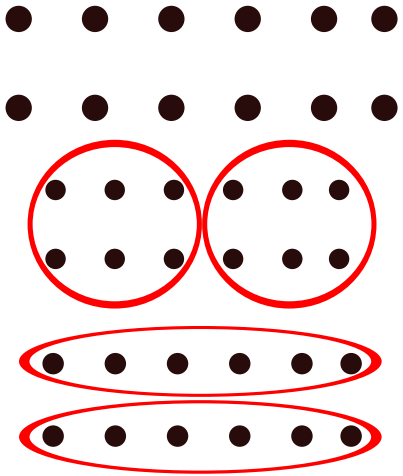
   Dimensionality reduction Find projections that carry important information.

   Compression Represent data using fewer bits.
3. Unsupervised learning is like supervised learning with missing outputs (or with missing inputs).

1. Given data

$$X = \{x_1, x_2, \ldots, x_m\}$$

learn to understand data, by re-representing it in some intelligent way.



Clustering

1. Given data set $X = \{x_1, x_2, \ldots, x_m\}$
2. Given distance function $d : X \times X \to \mathbb{R}$
   - 2.1 $d(x, y) \geq 0$, $\forall x, y \in X$.
   - 2.2 $d(x, y) = 0$ iff $x = y$.
   - 2.3 $d(x, y) = d(y, x)$.
3. A clustering C is a partition of X.

Definition

3.1 A clustering function is a function F which given a data set X and a distance function d it returns a partition C of X. $F : (X, d) \to C$

3.2 A clustering quality function is any function Q which given a data set X, a partioning C of X and a distance function d it returns a real number. $Q : (X, d, C) \to \mathbb{R}$

4. Given Q we can define F as $F(X, d) = \underset{C}{\text{argmax}}\ Q(X, d, C)$

K-means clustering

1. Let $\mu_1, \ldots, \mu_K \in \mathbb{R}^n$ be K cluster centroids. Then K-means clustering objective function is

$$J(X, d, C) = \sum_{k=1}^{K} \sum_{x \in C_k} d(x, \mu_k)^2$$

   $X \subset X'$ (e.g., data points are in $\mathbb{R}^n$)

2. The goal is to find a clustering that minimizes the objective function.

$$F(X, d) = \underset{\mu_1, \ldots, \mu_K \in X'}{\arg\min} \; J(X, d, C)$$

Theorem

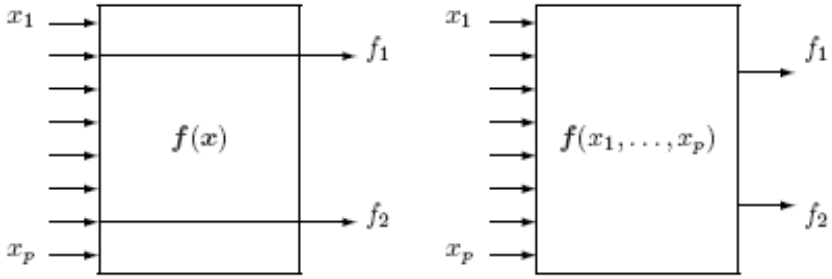
K-means clustering belongs to class of NP-Hard problems.

3. A good practical heuristic is Lloyd's algorithm

1. There are two main methods for reducing the dimensionality of inputs

   1.1 Feature selection: These methods select d ($d < D$) dimensions out of D dimensions and $D - d$ other dimensions are discarded.

   1.2 Feature extraction: Find a new set of d ($d < D$) dimensions that are combinations of the original dimensions.



Principal component analysis

1. In PCA, transformation $z = w^T x$ is used to reduce dimensionality of x.
2. To find the best k-first principle components dimensional using PCA, we compute the eigenvalues of covariance matrix Σ of data.
3. The eigenvalues of Σ are sorted in decreasing order

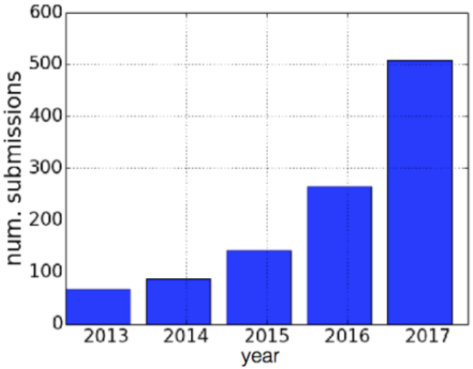
$$\lambda_1 \geq \lambda_2 \geq \ldots \lambda_{j-1} \geq \lambda_j \geq \ldots \geq \lambda_D \geq 0$$

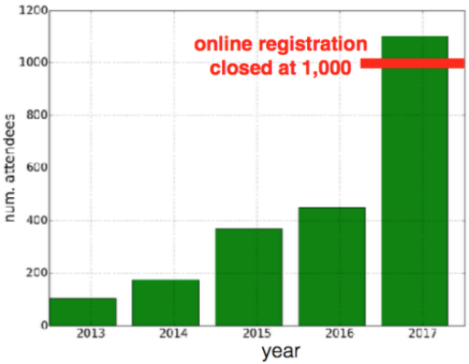
4. We then select the k largest eigenvalues, and their corresponding eigenvectors to form the best k-dimensional approximation.
5. Since Σ is symmetric, for two different eigenvalues, their corresponding eigenvectors are orthogonal.

Representation learning

Representation learning[1]

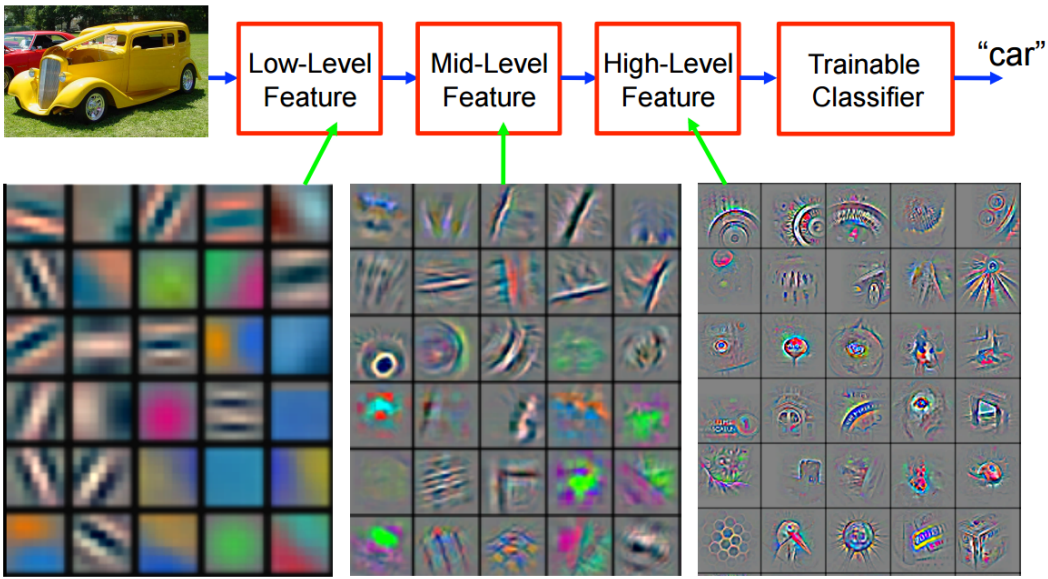
1. This type of learning is becoming a extremely popular topic.
2. Number of submissions at the International Conference on Learning Representations' (ICLR)





[1] This slide borrowed from Andre Martins slides.

1. As an example,



[2] This slide borrowed from Andre Martins slides.

References

Readings

1. Chapter 5 of Deep Learning Book[3]

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. The MIT Press.

References

- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. The MIT Press.
- Mitchell, Tom M. (1997). *Machine learning*. McGraw-Hill.
- Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar (2018). *Foundations of Machine Learning*. 2nd ed. The MIT Press.

Questions?