Title: Feature learning

URL: https://en.wikipedia.org/wiki/Feature_learning

PageID: 38870173

Categories: Category:Machine learning

Supervised learning

Unsupervised learning

Semi-supervised learning

Self-supervised learning

Reinforcement learning

Meta-learning

Online learning

Batch learning

Curriculum learning

Rule-based learning

Neuro-symbolic AI

Neuromorphic engineering

Quantum machine learning

Classification

Generative modeling

Regression

Clustering

Dimensionality reduction

Density estimation

Anomaly detection

Data cleaning

AutoML

Association rules

Semantic analysis

Structured prediction

Feature engineering

Feature learning

Learning to rank

Grammar induction

Ontology learning

Multimodal learning

Apprenticeship learning

Decision trees

Ensembles Bagging Boosting Random forest

Bagging

Boosting

Random forest

k -NN

Linear regression

Naive Bayes

Artificial neural networks

Logistic regression

Perceptron

Relevance vector machine (RVM)

Support vector machine (SVM)

BIRCH

CURE

Hierarchical

k -means

Fuzzy

Expectation–maximization (EM)

DBSCAN

OPTICS

Mean shift

Factor analysis

CCA

ICA

LDA

NMF

PCA

PGD

t-SNE

SDL

Graphical models Bayes net Conditional random field Hidden Markov

Bayes net

Conditional random field

Hidden Markov

RANSAC

k -NN

Local outlier factor

Isolation forest

Autoencoder

Deep learning

Feedforward neural network

Recurrent neural network LSTM GRU ESN reservoir computing

LSTM

GRU

ESN

reservoir computing

Boltzmann machine Restricted

Restricted

GAN

Diffusion model

SOM

Convolutional neural network U-Net LeNet AlexNet DeepDream

U-Net

LeNet

AlexNet

DeepDream

Neural field Neural radiance field Physics-informed neural networks

Neural radiance field

Physics-informed neural networks

Transformer Vision

Vision

Mamba

Spiking neural network

Memtransistor

Electrochemical RAM (ECRAM)

Q-learning

Policy gradient

SARSA

Temporal difference (TD)

Multi-agent Self-play

Self-play

Active learning

Crowdsourcing

Human-in-the-loop

Mechanistic interpretability

RLHF

Coefficient of determination

Confusion matrix

Learning curve

ROC curve

Kernel machines

Bias–variance tradeoff

Computational learning theory

Empirical risk minimization

Occam learning

PAC learning

Statistical learning

VC theory

Topological deep learning

AAAI

ECML PKDD

NeurIPS

ICML

ICLR

IJCAI

ML

JMLR

Glossary of artificial intelligence

List of datasets for machine-learning research List of datasets in computer vision and image processing

List of datasets in computer vision and image processing

Outline of machine learning

v

t

e

In machine learning (ML), feature learning or representation learning is a set of techniques that allow a system to automatically discover the representations needed for feature detection or classification from raw data. This replaces manual feature engineering and allows a machine to both learn the features and use them to perform a specific task.

Feature learning is motivated by the fact that ML tasks such as classification often require input that is mathematically and computationally convenient to process. However, real-world data, such as image, video, and sensor data, have not yielded to attempts to algorithmically define specific features. An alternative is to discover such features or representations through examination, without relying on explicit algorithms.

Feature learning can be either supervised, unsupervised, or self-supervised:

In supervised feature learning , features are learned using labeled input data. Labeled data includes input-label pairs where the input is given to the model, and it must produce the ground truth label as the output. This can be leveraged to generate feature representations with the model which result in high label prediction accuracy. Examples include supervised neural networks , multilayer perceptrons , and dictionary learning .

In unsupervised feature learning , features are learned with unlabeled input data by analyzing the relationship between points in the dataset. Examples include dictionary learning, independent component analysis , matrix factorization , and various forms of clustering .

In self-supervised feature learning , features are learned using unlabeled data like unsupervised learning, however input-label pairs are constructed from each data point, enabling learning the structure of the data through supervised methods such as gradient descent . Classical examples include word embeddings and autoencoders . Self-supervised learning has since been applied to many modalities through the use of deep neural network architectures such as convolutional neural networks and transformers .

Supervised

Supervised feature learning is learning features from labeled data. The data label allows the system to compute an error term, the degree to which the system fails to produce the label, which can then be used as feedback to correct the learning process (reduce/minimize the error). Approaches include:

Supervised dictionary learning

Dictionary learning develops a set (dictionary) of representative elements from the input data such that each data point can be represented as a weighted sum of the representative elements. The dictionary elements and the weights may be found by minimizing the average representation error (over the input data), together with L1 regularization on the weights to enable sparsity (i.e., the representation of each data point has only a few nonzero weights).

Supervised dictionary learning exploits both the structure underlying the input data and the labels for optimizing the dictionary elements. For example, this supervised dictionary learning technique applies dictionary learning on classification problems by jointly optimizing the dictionary elements, weights for representing data points, and parameters of the classifier based on the input data. In particular, a minimization problem is formulated, where the objective function consists of the classification error, the representation error, an L1 regularization on the representing weights for each data point (to enable sparse representation of data), and an L2 regularization on the parameters of the classifier.

Neural networks

Neural networks are a family of learning algorithms that use a "network" consisting of multiple layers of inter-connected nodes. It is inspired by the animal nervous system, where the nodes are viewed as neurons and edges are viewed as synapses. Each edge has an associated weight, and the network defines computational rules for passing input data from the network's input layer to the output layer. A network function associated with a neural network characterizes the relationship between input and output layers, which is parameterized by the weights. With appropriately defined network functions, various learning tasks can be performed by minimizing a cost function over the network function (weights).

Multilayer neural networks can be used to perform feature learning, since they learn a representation of their input at the hidden layer(s) which is subsequently used for classification or regression at the output layer. The most popular network architecture of this type is Siamese networks .

Unsupervised

Unsupervised feature learning is learning features from unlabeled data. The goal of unsupervised feature learning is often to discover low-dimensional features that capture some structure underlying the high-dimensional input data. When the feature learning is performed in an unsupervised way, it enables a form of semisupervised learning where features learned from an unlabeled dataset are then employed to improve performance in a supervised setting with labeled data. Several approaches are introduced in the following.

K -means clustering

K -means clustering is an approach for vector quantization. In particular, given a set of n vectors, k -means clustering groups them into k clusters (i.e., subsets) in such a way that each vector belongs to the cluster with the closest mean. The problem is computationally NP-hard , although suboptimal greedy algorithms have been developed.

K-means clustering can be used to group an unlabeled set of inputs into k clusters, and then use the centroids of these clusters to produce features. These features can be produced in several ways. The simplest is to add k binary features to each sample, where each feature j has value one iff the j th centroid learned by k -means is the closest to the sample under consideration. It is also possible to use the distances to the clusters as features, perhaps after transforming them through a radial basis function (a technique that has been used to train RBF networks). Coates and Ng note that certain variants of k -means behave similarly to sparse coding algorithms.

In a comparative evaluation of unsupervised feature learning methods, Coates, Lee and Ng found that k -means clustering with an appropriate transformation outperforms the more recently invented auto-encoders and RBMs on an image classification task. K -means also improves performance in the domain of NLP , specifically for named-entity recognition ; there, it competes with Brown clustering , as well as with distributed word representations (also known as neural word embeddings).

Principal component analysis

Principal component analysis (PCA) is often used for dimension reduction. Given an unlabeled set of n input data vectors, PCA generates p (which is much smaller than the dimension of the input data) right singular vectors corresponding to the p largest singular values of the data matrix, where the k th row of the data matrix is the k th input data vector shifted by the sample mean of the input (i.e., subtracting the sample mean from the data vector). Equivalently, these singular vectors are the eigenvectors corresponding to the p largest eigenvalues of the sample covariance matrix of the input vectors. These p singular vectors are the feature vectors learned from the input data, and they represent directions along which the data has the largest variations.

PCA is a linear feature learning approach since the p singular vectors are linear functions of the data matrix. The singular vectors can be generated via a simple algorithm with p iterations. In the i th iteration, the projection of the data matrix on the (i-1) th eigenvector is subtracted, and the i th singular vector is found as the right singular vector corresponding to the largest singular of the residual data matrix.

PCA has several limitations. First, it assumes that the directions with large variance are of most interest, which may not be the case. PCA only relies on orthogonal transformations of the original data, and it exploits only the first- and second-order moments of the data, which may not well characterize the data distribution. Furthermore, PCA can effectively reduce dimension only when the input data vectors are correlated (which results in a few dominant eigenvalues).

Local linear embedding

Local linear embedding (LLE) is a nonlinear learning approach for generating low-dimensional neighbor-preserving representations from (unlabeled) high-dimension input. The approach was proposed by Roweis and Saul (2000). The general idea of LLE is to reconstruct the original high-dimensional data using lower-dimensional points while maintaining some geometric properties of the neighborhoods in the original data set.

LLE consists of two major steps. The first step is for "neighbor-preserving", where each input data point Xi is reconstructed as a weighted sum of K nearest neighbor data points, and the optimal weights are found by minimizing the average squared reconstruction error (i.e., difference between an input point and its reconstruction) under the constraint that the weights associated with each point sum up to one. The second step is for "dimension reduction," by looking for vectors in a lower-dimensional space that minimizes the representation error using the optimized weights in the first step. Note that in the first step, the weights are optimized with fixed data, which can be solved as a least squares problem. In the second step, lower-dimensional points are optimized with fixed weights, which can be solved via sparse eigenvalue decomposition.

The reconstruction weights obtained in the first step capture the "intrinsic geometric properties" of a neighborhood in the input data. It is assumed that original data lie on a smooth lower-dimensional manifold , and the "intrinsic geometric properties" captured by the weights of the original data are also expected to be on the manifold. This is why the same weights are used in the second step of LLE. Compared with PCA, LLE is more powerful in exploiting the underlying data structure.

Independent component analysis

Independent component analysis (ICA) is a technique for forming a data representation using a weighted sum of independent non-Gaussian components. The assumption of non-Gaussian is imposed since the weights cannot be uniquely determined when all the components follow Gaussian distribution.

Unsupervised dictionary learning

Unsupervised dictionary learning does not utilize data labels and exploits the structure underlying the data for optimizing dictionary elements. An example of unsupervised dictionary learning is sparse coding , which aims to learn basis functions (dictionary elements) for data representation from unlabeled input data. Sparse coding can be applied to learn overcomplete dictionaries, where the number of dictionary elements is larger than the dimension of the input data. Aharon et al. proposed algorithm K-SVD for learning a dictionary of elements that enables sparse representation.

Multilayer/deep architectures

The hierarchical architecture of the biological neural system inspires deep learning architectures for feature learning by stacking multiple layers of learning nodes. These architectures are often designed based on the assumption of distributed representation : observed data is generated by the interactions of many different factors on multiple levels. In a deep learning architecture, the output of each intermediate layer can be viewed as a representation of the original input data. Each level uses the representation produced by the previous, lower level as input, and produces new representations as output, which are then fed to higher levels. The input at the bottom layer is raw data, and the output of the final, highest layer is the final low-dimensional feature or representation.

Restricted Boltzmann machine

Restricted Boltzmann machines (RBMs) are often used as a building block for multilayer learning architectures. An RBM can be represented by an undirected bipartite graph consisting of a group of binary hidden variables , a group of visible variables, and edges connecting the hidden and visible nodes. It is a special case of the more general Boltzmann machines with the constraint of no intra-node connections. Each edge in an RBM is associated with a weight. The weights together with the connections define an energy function , based on which a joint distribution of visible and hidden nodes can be devised. Based on the topology of the RBM, the hidden (visible) variables are independent, conditioned on the visible (hidden) variables. [clarification needed] Such conditional independence facilitates computations.

An RBM can be viewed as a single layer architecture for unsupervised feature learning. In particular, the visible variables correspond to input data, and the hidden variables correspond to feature detectors. The weights can be trained by maximizing the probability of visible variables using Hinton 's contrastive divergence (CD) algorithm.

In general, training RBMs by solving the maximization problem tends to result in non-sparse representations. Sparse RBM was proposed to enable sparse representations. The idea is to add a regularization term in the objective function of data likelihood, which penalizes the deviation of the expected hidden variables from a small constant p {\displaystyle p} . RBMs have also been used to obtain disentangled representations of data, where interesting features map to separate hidden units.

Autoencoder

An autoencoder consisting of an encoder and a decoder is a paradigm for deep learning architectures. An example is provided by Hinton and Salakhutdinov where the encoder uses raw data (e.g., image) as input and produces feature or representation as output and the decoder uses the extracted feature from the encoder as input and reconstructs the original input raw data as output. The encoder and decoder are constructed by stacking multiple layers of RBMs. The parameters involved in the architecture were originally trained in a greedy layer-by-layer manner: after one layer of feature detectors is learned, they are fed up as visible variables for training the corresponding RBM. Current approaches typically apply end-to-end training with stochastic gradient descent methods. Training can be repeated until some stopping criteria are satisfied.

Self-supervised

Self-supervised representation learning is learning features by training on the structure of unlabeled data rather than relying on explicit labels for an information signal . This approach has enabled the combined use of deep neural network architectures and larger unlabeled datasets to produce deep feature representations. Training tasks typically fall under the classes of either contrastive, generative or both. Contrastive representation learning trains representations for associated data pairs, called positive samples, to be aligned, while pairs with no relation, called negative samples, are contrasted. A larger portion of negative samples is typically necessary in order to prevent catastrophic collapse, which is when all inputs are mapped to the same representation. Generative representation learning tasks the model with producing the correct data to either match a restricted input or reconstruct the full input from a lower dimensional representation.

A common setup for self-supervised representation learning of a certain data type (e.g. text, image, audio, video) is to pretrain the model using large datasets of general context, unlabeled data. Depending on the context, the result of this is either a set of representations for common data segments (e.g. words) which new data can be broken into, or a neural network able to convert each new data point (e.g. image) into a set of lower dimensional features. In either case, the output representations can then be used as an initialization in many different problem settings where labeled data may be limited. Specialization of the model to specific tasks is typically done with supervised learning, either by fine-tuning the model / representations with the labels as the signal, or freezing the representations and training an additional model which takes them as an input.

Many self-supervised training schemes have been developed for use in representation learning of various modalities , often first showing successful application in text or image before being transferred to other data types.

Text

Word2vec is a word embedding technique which learns to represent words through self-supervision over each word and its neighboring words in a sliding window across a large corpus of text. The model has two possible training schemes to produce word vector representations, one generative and one contrastive. The first is word prediction given each of the neighboring words as an input. The second is training on the representation similarity for neighboring words and representation dissimilarity for random pairs of words. A limitation of word2vec is that only the pairwise co-occurrence structure of the data is used, and not the ordering or entire set of context words. More recent transformer-based representation learning approaches attempt to solve this with word prediction tasks. GPTs pretrain on next word prediction using prior input words as context, whereas BERT masks random tokens in order to provide bidirectional context.

Other self-supervised techniques extend word embeddings by finding representations for larger text structures such as sentences or paragraphs in the input data. Doc2vec extends the generative training approach in word2vec by adding an additional input to the word prediction task based on the paragraph it is within, and is therefore intended to represent paragraph level context.

Image

The domain of image representation learning has employed many different self-supervised training techniques, including transformation, inpainting, patch discrimination and clustering.

Examples of generative approaches are Context Encoders, which trains an AlexNet CNN architecture to generate a removed image region given the masked image as input, and iGPT, which applies the GPT-2 language model architecture to images by training on pixel prediction after reducing the image resolution .

Many other self-supervised methods use siamese networks , which generate different views of the image through various augmentations that are then aligned to have similar representations. The challenge is avoiding collapsing solutions where the model encodes all images to the same representation. SimCLR is a contrastive approach which uses negative examples in order to generate image representations with a ResNet CNN . Bootstrap Your Own Latent (BYOL) removes the need for negative samples by encoding one of the views with a slow moving average of the model parameters as they are being modified during training.

Graph

The goal of many graph representation learning techniques is to produce an embedded representation of each node based on the overall network topology . node2vec extends the word2vec training technique to nodes in a graph by using co-occurrence in random walks through the graph as the measure of association. Another approach is to maximize mutual information , a measure of similarity, between the representations of associated structures within the graph. An example is Deep Graph Infomax, which uses contrastive self-supervision based on mutual information between the representation of a "patch" around each node, and a summary representation of the entire graph. Negative samples are obtained by pairing the graph representation with either representations from another graph in a multigraph training setting, or corrupted patch representations in single graph training.

Video

With analogous results in masked prediction and clustering, video representation learning approaches are often similar to image techniques but must utilize the temporal sequence of video frames as an additional learned structure. Examples include VCP, which masks video clips and trains to choose the correct one given a set of clip options, and Xu et al., who train a 3D-CNN to identify the original order given a shuffled set of video clips.

Audio

Self-supervised representation techniques have also been applied to many audio data formats, particularly for speech processing . Wav2vec 2.0 discretizes the audio waveform into timesteps via temporal convolutions , and then trains a transformer on masked prediction of random timesteps using a contrastive loss. This is similar to the BERT language model , except as in many SSL approaches to video, the model chooses among a set of options rather than over the entire word vocabulary.

Multimodal

Self-supervised learning has also been used to develop joint representations of multiple data types. Approaches usually rely on some natural or human-derived association between the modalities as an implicit label, for instance video clips of animals or objects with characteristic sounds, or captions written to describe images. CLIP produces a joint image-text representation space by training to align image and text encodings from a large dataset of image-caption pairs using a contrastive loss. MERLOT Reserve trains a transformer-based encoder to jointly represent audio, subtitles and video frames from a large dataset of videos through 3 joint pretraining tasks: contrastive masked prediction of either audio or text segments given the video frames and surrounding audio and text context, along with contrastive alignment of video frames with their corresponding captions.

Multimodal representation models are typically unable to assume direct correspondence of representations in the different modalities, since the precise alignment can often be noisy or ambiguous. For example, the text "dog" could be paired with many different pictures of dogs, and correspondingly a picture of a dog could be captioned with varying degrees of specificity. This limitation means that downstream tasks may require an additional generative mapping network between modalities to achieve optimal performance, such as in DALLE-2 for text to image generation.

Dynamic Representation Learning

Dynamic representation learning methods generate latent embeddings for dynamic systems such as dynamic networks. Since particular distance functions are invariant under particular linear transformations, different sets of embedding vectors can actually represent the same/similar information. Therefore, for a dynamic system, a temporal difference in its embeddings may be explained by misalignment of embeddings due to arbitrary transformations and/or actual changes in the system. Therefore, generally speaking, temporal embeddings learned via dynamic representation learning methods should be inspected for any spurious changes and be aligned before consequent dynamic analyses.

See also

Automated machine learning (AutoML)

Deep learning

Geometric feature learning

Feature detection (computer vision)

Feature extraction

Word embedding

Vector quantization

Variational autoencoder

References