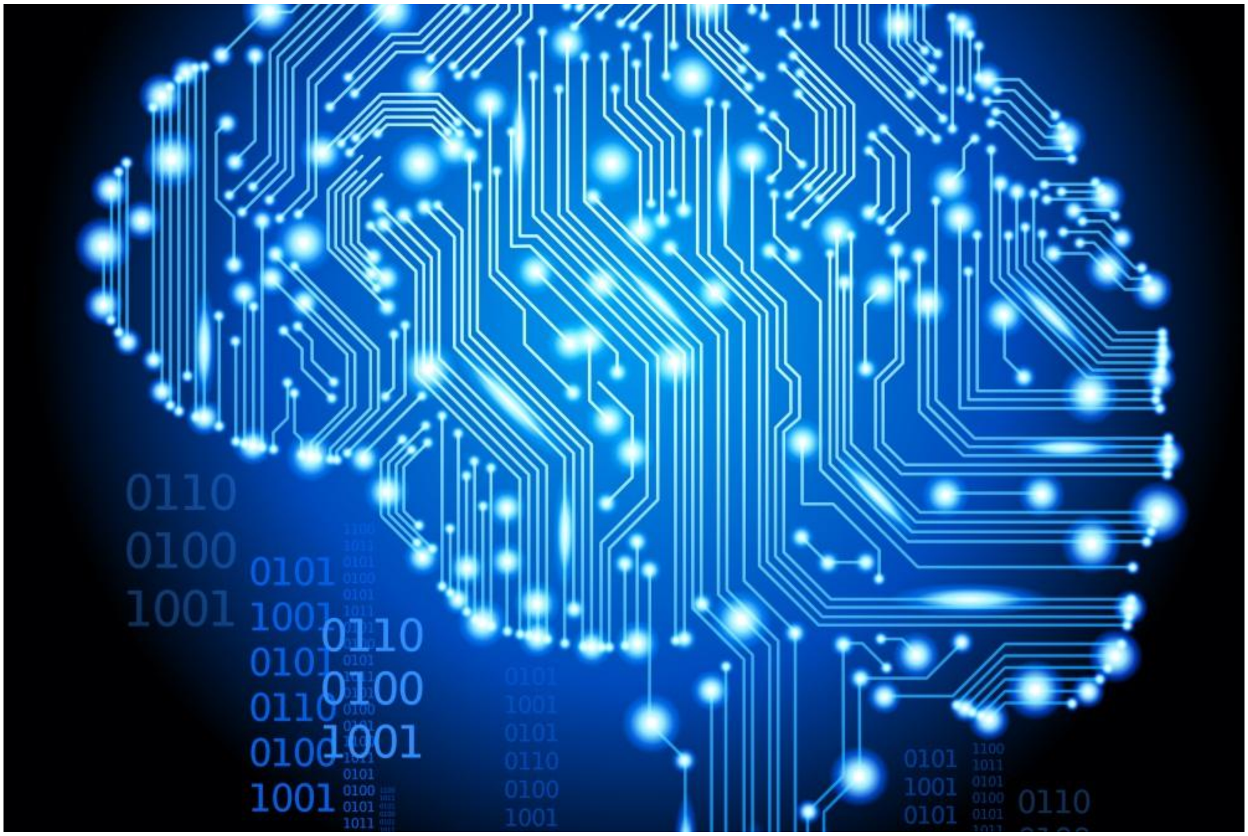


Towards Learning based Reasoning Systems

Hongzheng Wang, Lang Gao, Chuanzhu Xu and Cong Rao

Department of Computer and Information Sciences
Temple University
Philadelphia, PA 19122

Abstract

In this project, we will explore ways to automatically generate explicit and implicit rules in a reasoning system. Considering the resemblance of a search tree and a decision tree, the explicit rules may be obtained by decision tree learning. Inspired by the recent success of Deep Neural Network, we also would like to apply related methods for learning the implicit rules. The implemented systems will be evaluated in various applications and on different datasets.

Introduction

Decision Tree

In traditional rule-based reasoning systems, the rules are often designed manually, typically by people with domain-specific knowledge. Often a large quantity of rules have to be specified in order to solve a problem. Besides, the manually designed rules could be inaccurate and incomplete, as humans will get tired and sometimes make mistakes. Thus, an automatic way to design the rules is necessary to save human labor.

Compared to traditional rule-based reasoning systems, building explicit rules via learning decision tree [25] is more accurate and time efficient. The decision tree makes connections between classes and data attributes in a manner of top-down problem solving. A decision tree is simple way to represent how the data can be divided. Each internal node in the tree tests one of the data attributes and each leaf node of the tree corresponds to one class. Thus, it is very easy to understand how decisions are made in each step. It is analogous to the conscious reasoning process of human beings.

In this project, we test the method in the context of classifying the Iris flowers. The performance is compared to the classic logic based method (or First-order Predicate Calculus), where the rules have to be designed manually. Some other advanced supervised learning methods are also added for comparison.

Deep Neural Network

In practice, it is sometimes difficult (or at least not efficient) for us to design explicit rules to solve certain problems, such as learning a language, playing Go or recognizing a face. For human beings, these tasks are often successfully accomplished by applying unconscious reasoning (or intuitions). Thus, it would be interesting to explore the way of programming those intuitions in computers.

The model of Neural Network is an early attempt to simulate the mechanism of human brains. It dates back to the mathematical model of a single neuron [1]. Later on, networks with a few layers are developed and successfully applied to a lot of problems in practice. The backpropagation algorithm proposed by Rumelhart et al. [2] is often used for learning a Neural Network. More recently, people find that the power of the neural networks can be greatly improved by just making them deeper. These networks are often called the Deep Neural Networks (DNN) and related learning methods are mentioned as Deep Learning.

Current advance in computer hardware has also greatly facilitated the development and applications of Deep Learning. And now DNN has been widely used in various fields including Speech Recognition [3], Natural Language Processing such as Machine Translation [4] and Language Modeling [5], Drug Detection [6] and Toxicology [7], and Computer Vision [8].

In this project, we exploit DNN for measuring the image similarities. It is expected that the learned similarity metric would be consistent with the human intuitions. The performance is evaluated on several popular dataset such as MNIST [20], CIFAR-10 [21] and PASCAL VOC 2007 [22].

Emotion and Creativity

In addition to learning for problem solving, we also would like to investigate the way to build intelligence for emotion and creativity, as they are important traits of human beings. Similar problems have also been raised in the Turing test, where it requires AI to exhibit intelligent behavior indistinguishable from that of a human. Such behavior is definitely more than logical reasoning, as emotion and creativity are also important traits of human beings. In this project, we conduct a short survey on these two topics, serving as a high level thinking for the project as well as the AI course.

Emotion

In the traditional view, people believe that feeling and thinking is distinct from each other and emotion will interfere with rational thinking. However, studies in recent years showed that emotion plays an important role as they have helped cognitive psychology in many aspects. In fact, feeling or emotion are important guide one’s behavior [9]. A study by Brian Knutson demonstrated that activation of distinct neural circuits related to anticipatory affect precedes and supports consumers’ purchasing decisions [10]. Another study by Joshua Greene showed that emotions have a significant impact on the ordinary moral decision-making [11]. Since intelligence is related to emotion, the ability to feel and express emotion as human should also be a focus of AI.

So can machines have feelings in addition to the capability of thinking? One feasible solution could be imitating the senses of human. In emotion simulation, the physical interaction with the environment is a crucial factor. If someone cannot perceive the danger nearby, it is not possible to stimulate the feeling of fear. We can attach a series of sensors to machines, thus they are able to see, listen, etc. Then the information gathered by these sensors can be further processed by other AI algorithms to simulate emotions of human.

However, currently machines can not completely simulate all the emotions of human beings. It is mainly because that machines do not have a biological body as human. For human beings, many emotions come with activity of hormones, such as thirst and hunger. A machine may feel cold or hot by temperature sensors, but it does not have a digestive system or the requirement for food. Hence it is not natural to simulate such emotions on a machine. Besides, some emotions can influence human's health. For example, tension and anxiety may lead to insomnia. It becomes a question if these processes are necessary to be simulated.

Intuition and instinct are also the traits that machines could hardly possess. Intuition is the ability to understand something immediately, without the need for conscious reasoning or study. And thinking without reasoning is now inconceivable for most AI algorithms. Instinct is a natural tendency to behave in a particular way or a natural ability to know something which is not learned. It comes from the evolution process of creature and machines will of course not have such ability since it cannot evolve like a creature. Since machines lack these important traits, they may not faithfully experience certain related feelings, such as the maternal instincts towards offspring, the love to opposite sex and the gut feelings from intuition.

Creativity

Creativity is also considered as an inexplicable aspect of human behaviors. As an essential part of human intelligence that fosters the development of human society, it is commonly acknowledged as an important topic of artificial intelligence [12]. However, there are criticisms denying the possibility that a machine can ever become creative because no rules or procedures can be expressed.

Commonly recognized features of creativity are novelty, unconventionality, usefulness, valuableness, flexibility, etc. However, defining the criteria for accurately describing creativity is still a major challenge for its simulation. If we can come up with an algorithmic definition of creativity, then creativity can be achieved by defining problems in a way so as to maximize the chance of discovering the rules and procedures underlying human behavior, as suggested by R Kurzweil et al. [13].

Since creativity can be reasoned as discovering hidden solutions that are not easily discoverable with conventional approaches, creativity simulation can be implemented as searching through the solution space and ruling out those solutions that are implausible or inappropriate.

Related Work

Decision Tree

There are two main types of decision tree, namely the classification tree and the regression tree. Currently popular decision tree learning algorithms include ID3, C4.5 [24] and Random Forest [26]. C4.5 builds the decision tree in the same way as ID3 [25]. For each node of the tree, C4.5 chooses an attribute that can divide the data into subsets that satisfy certain optimal criterion for selection. Compared to ID3, the C4.5 algorithm can handle continuous or discrete attributes as well as missing attribute values. The

C4.5 uses the post-pruning strategy to limit the size of a decision tree. The ID3 algorithm is a predecessor to the C4.5 algorithm. Compared to C4.5, ID3 does not guarantee an optimal solution and it sometimes may get stuck in local optima. Also ID3 is not suitable for processing continuous data. Random Forest is another algorithm which consists of an ensemble of decision trees. It takes the majority votes of the involved trees for classification. Compared to C4.5, Random Forest is able to handle unbalanced and missing data. The weaknesses of random forest is that when it is applied to the task of regression, it may not perform very well on the unseen test data.

Deep Neural Network

The techniques of DNN have been applied in various applications. The first industrial application of DNN is the phonetic classification for speech recognition [14]. Traditional speech recognition systems use Gaussian mixture models (GMM) to evaluate the fitness of a hidden Markov model, which deals with the temporal variability of speech. When DNN was first tested for this application, it showed significant improvement over GMM even without optimizing the types of hidden units and networks architectures.

The adoption of DNN on image classification is accelerated by the discovery made in the University of Toronto [15]. With a DNN consisted of five convolutional layers, Alex Krizhevsky et al. achieved record-breaking error rate when applied to classify millions of high-resolution images. This work is considered as a major breakthrough recently in the field of Computer Vision.

The popularity of DNN even attracted chemists to apply it for drug detection. For example, Junshui et al. [27] demonstrated that DNN shows superior performance over traditional methods like Random Forest [26] in predicting activities of drug molecules. Likewise, researchers in bioinformatics also find useful applications of DNN. Hui et al. [28] built a computational model using DNN to predict splicing in human tissues, and provided useful insights in studying diseases and disorders.

Emotion and Creativity

Rapid progress has been made in the current research of emotion, such as reading emotions of a human, which is a starting point of AI emotion simulation. To a certain extent, machines already have the ability to recognize human emotions by scanning facial features [16]. The Computer Expression Recognition Toolbox (CERT) [17] toolkit, for example, can track facial action units described by “Facial Action Coding System” (FACS) [18]. CERT is able to automatically encode the intensity of 19 different facial actions from the FACS and 6 different prototypical facial expressions, and achieves a high accuracy on the recognition test.

One of the earliest examples of creative programs is AM [29], in which the authors formulate the creativity as learning new heuristics via discovery. The program explores in the space of mathematical concepts and is able to find interesting concepts such as prime numbers. Bacon [30] is another famous creative program. It is capable of rediscovering many important physical laws, such as Kepler's third law, Ohm's law, etc. Although the program suffers from many limitations including capability of deriving laws from existing theories, robustness against noisy data, it makes us more confident that scientific discoveries can be understood just like natural phenomenons that can be explained with rules.

It’s safe to argue that machines may possess creativity and emotion someday in the future. Some creative AI systems can now convey their feelings via art works [32] which generally carries emotions. Music composition has already been achieved. The project led by Eduardo Miranda and Slawomir Nasuto [31] composed music with a computer by monitoring a person’s emotional response to sounds. Researchers played music pieces to test the subjects as they underwent magnetic resonance imaging (MRI) scans while the program monitored their brains for emotional indicators. Then computer generated a new music piece designed to create a similar emotional response. This experiment showed that AI is able to express emotions in the form of music.

Likewise, researchers also demonstrated the possibility of using AI based “digit painters” to create art paintings. Sara et al. [32] proposed ePainterly, which is a creative module that analyzes the emotion spaces from photography, and then uses them for unique aesthetic visualization. Another example is the famous robotic artist, Paul, developed by Patrick et al. [33]. It is capable of making portraits based on what it sees. It imitates the drawing skills and styles of a human artist even though it does not learn from experience, and the output also has an emotional and aesthetic effect.

Approach

Learning Decision Tree

Given a database

$$D \quad = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\},$$

where N is the number of data instances, x_i are the attributes (typically represented a vector) of a data instance, y_i is the class label of x_i, a decision tree can be then induced following the framework in the Hunt’s Algorithm.

Hunt’s Algorithm

Let D_t be the set of training records that reach a node t,

1) If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t;
2) If D_t is an empty set, then t is a leaf node labeled by the default class y_d;
3) If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets.

We can grow a full tree by recursively applying the procedure to the subset in each tree node. The tree induction in the procedure 3) uses a greedy strategy, which splits the records based on an attribute test that optimizes certain criterion.

Decision Tree Induction

To grow a decision tree, we need to split the data recursively until the termination condition is reached. For a tree node t, the best split is determined by checking the impurity before and after the split. The split with a lower overall impurity is prefered. The following are some frequently used impurity measures

1) GINI Index

$$GINI(t) = 1 - \sum_{i=1}^{c} \quad p(i|t)^2,$$

2) Entropy

$$Entropy(t) = -\sum_{i=1}^{c} \quad p(\,i|t)\,log\,p(i|t),$$

3) Classification Error

$$Error(t) = 1 - \max_{i} p(i|t),$$

where c is the number of classes and $p(i|t)$ is the frequency of class i in node t.

Suppose a set of new nodes $\{t_1, t_2, \ldots, t_m\}$ is obtained after the split, and each node contains n_i data instances. The total number of data instances before the split is n. Then the reduction of impurity with respect to some measure $I(t)$ is computed as

$$I_{reduction} = I(t) - \sum_{i=1}^{m} \quad \frac{n_i}{n} I(t_i).$$

Then, we can select the split with the maximum reduction in terms of impurity.

In the method C4.5, the Entropy is used to measure the impurity. Besides, the intrinsic information of a split is also considered for selecting the attribute, which is defined as

$$I_{intrinsic} = -\sum_{i=1}^{m} \quad \frac{n_i}{n} log \frac{n_i}{n}.$$

It tells us how much information do we need to tell which branch an instance belongs to. Attributes with higher intrinsic information are less useful. To select the split with the highest reduction in impurity and the lowest intrinsic information, the method C4.5 proceeds to use the Gain Ratio as the evaluation metric, which is defined as

$$Gain\ Ratio = \frac{I_{reduction}}{I_{intrinsic}}.$$

It reduces its bias towards multi-values attributes and takes the number and size of branches into account when choosing an attribute.

Decision Tree Pruning

Recursively applying the the decision tree induction may end up with a tree that is infinitely large. Thus, we have to employ certain strategy to limit the size of the decision tree, as a large decision tree may bring about the concern of complexity in both time and space. Also, there may be also potential issue of the overfitting problem, when the model performs very well in the training data but very poorly in the unseen test data.

1) Pre-pruning (or Early Termination)
    a) Stop if it reaches the maximum depth or maximum number of leaf nodes
    b) Stop if all instances belong to the same class
    c) Stop if all the attribute values are the same
    d) Stop if number of instances is less than some user-specified threshold
    e) Stop if expanding the current node does not improve impurity measures
2) Post-pruning
    a) Grow a decision tree with a maximum depth
    b) Trim the nodes of the decision tree in a bottom-up fashion
    c) If classification error improves after trimming, replace subtree by a leaf node

In the method C4.5, the post-pruning strategy is applied in this process.

Deep Neural Network

In this project, we are to measure the similarity between two images by learning a Deep Neural Network. As we are focusing on the pairwise relationships between the images, the database we use is presented in the form

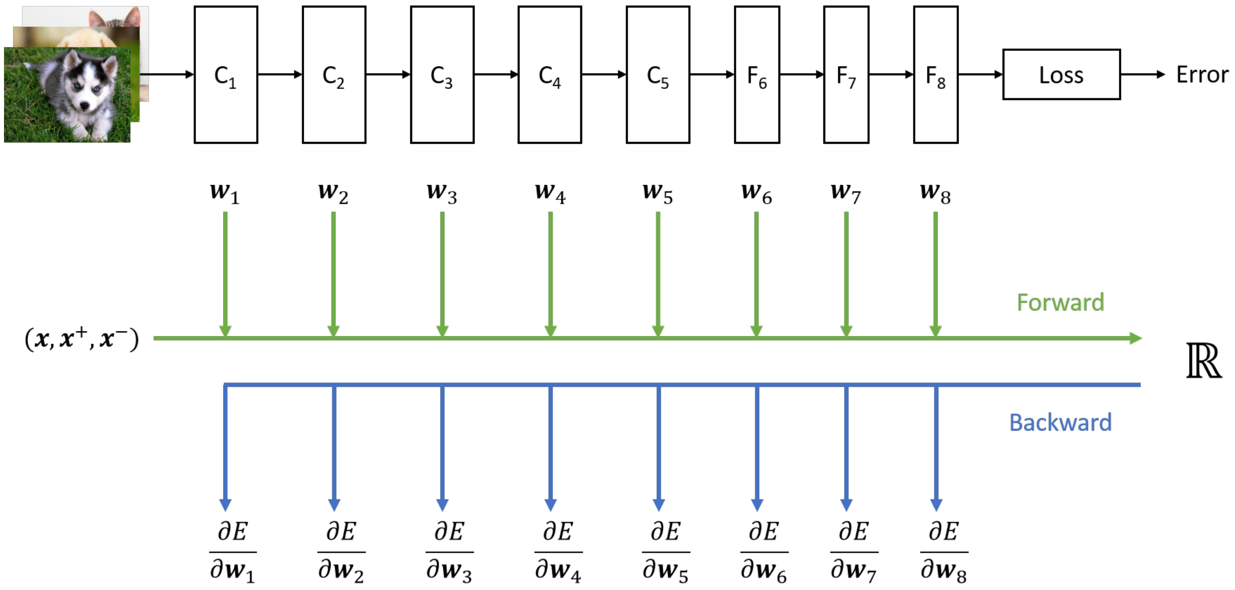
$$D \quad = \{(x_1, {x_1}^+, {x_1}^-), (x_2, {x_2}^+, {x_2}^-), \dots, (x_N, {x_N}^+, {x_N}^-)\},$$

where ${x_i}^+$ is an image relevant to the image x_i and ${x_i}^-$ is an image irrelevant to x_i. As we can see, each training example is a triplet of three images.

In traditional supervised learning, a database $D \quad = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ is often used in the training process, and most publicly available databases are stored in this form. In order to utilize the datasets constructed for traditional supervised learning, we can convert them to the desired format by making random combinations of the data instances. Namely, for each instance x_i, we obtain ${x_i}^+$ by randomly sample one instance from the same class. Similarly we generate ${x_i}^-$ by randomly sample one instance from the other class. Considering there are potentially $O(N^3)$ combinations, it can be used to enlarge certain dataset if the training examples are insufficient for learning a Deep Neural Network.

Learning Image Similarity

In the context of applying Deep Neural Network to image related applications, often we use a network structure called Convolutional Neural Network (CNN), which is specially designed for processing images. The following shows the standard process for training a CNN.


C_1
C_2
C_3
C_4
C_5
F_6
F_7
F_8
Loss
Error
$\boldsymbol{w}_1$
$\boldsymbol{w}_2$
$\boldsymbol{w}_3$
$\boldsymbol{w}_4$
$\boldsymbol{w}_5$
$\boldsymbol{w}_6$
$\boldsymbol{w}_7$
$\boldsymbol{w}_8$
Forward
$(\boldsymbol{x}, \boldsymbol{x}^+, \boldsymbol{x}^-)$
$\mathbb{R}$
Backward
$\frac{\partial E}{\partial \boldsymbol{w}_1}$
$\frac{\partial E}{\partial \boldsymbol{w}_2}$
$\frac{\partial E}{\partial \boldsymbol{w}_3}$
$\frac{\partial E}{\partial \boldsymbol{w}_4}$
$\frac{\partial E}{\partial \boldsymbol{w}_5}$
$\frac{\partial E}{\partial \boldsymbol{w}_6}$
$\frac{\partial E}{\partial \boldsymbol{w}_7}$
$\frac{\partial E}{\partial \boldsymbol{w}_8}$

An illustration for CNN learning in our framework

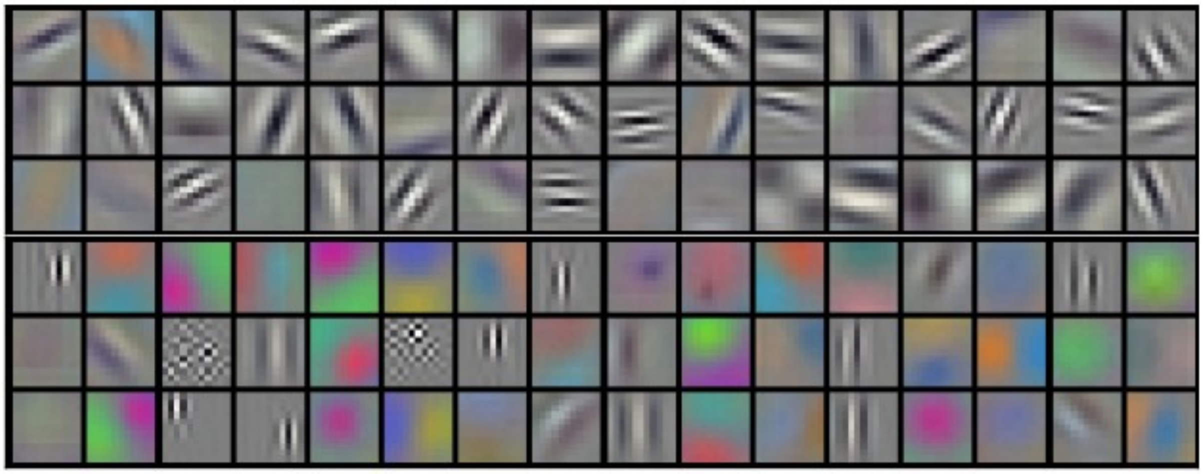
The figure shows a CNN with 8 layers, in which 5 are convolutional layers and 3 are fully connected layers. The parameters of each layer are denoted as w_i. There are two stages in the CNN training process, one is forward propagation, and the other one is back propagation. In the stage of forward propagation, the output of each layer is computed as well as a value indicating the overall training error. In the stage of backward propagation, the overall error is propagated to previous layers in terms of the partial derivatives with respect to each w_i. Then these parameters w can be updated using traditional methods such as Gradient Descent.

Forward Propagation

The goal of Forward Propagation is to compute the output of each layer in the CNN. Different from traditional Artificial Neural Network, typically we may observe the following types of layers in a CNN.

Convolutional Layer

The convolutional layer is the basic building block of a convolutional network. It contains a set of filters, in which each filter is a collection of neurons arranged in a 3D volume. During the forward propagation, each filter slides across the width and height of the input volume and produces a 2D activation map. Each entry in the activation map is the dot product between the local volume in the input and parameters in the filter when it slides.



An illustration of the parameters learned in a convolutional layer

The above figure shows a convolutional layer with 96 filters, where each filter is a 3D volume of size $11 \times 11 \times 3$. The parameters in each filter are learned from the training data and they are all visualized as color images of 3 channels.

Fully-connected Layer

Like the one used in traditional Artificial Neural Networks, neurons in a fully-connected layer are arranged in a 1D vector. They have full connections to all activations in the previous layer. The output of a fully-connected layer can be computed with a matrix multiplication followed by a bias offset.

The only difference between fully-connected layer and convolutional layer is that the neurons in the convolutional layer are connected only to a local region in the input, and that many of the neurons in a convolutional volume share parameters. However, the neurons in both kinds of layers still compute dot products, so their functionalities are identical.

Nonlinear Gating Layer

The convolutional layer and fully-connected layer essentially contain a set of linear filters, and often they are followed by some nonlinear gating function. Most frequently used nonlinear gating functions include Sigmoid Function, Rectified Linear Unit (ReLU) and Drop Out.

Pooling Layer

A Pooling layer is often inserted between consecutive convolutional layers in a CNN. It is used to reduce the spatial size of output feature map, thus the amount of parameters and computation in later stages are also reduced. It also helps alleviate the overfitting problem. Conventionally used pooling layers include Max-pooling and Sum-pooling.

Normalization Layer

Many types of normalization layers have been proposed for use in CNN architectures such as Batch Normalization [22]. Some of them are implemented as an inspiration from the biological observation in the human brain.

Backward Propagation

In Backward Propagation, the partial derivatives of the error with respect to the parameters are computed, and they are updated such that the network can adapt to the examples it currently see. In our problem formulation, the overall energy function or error function is defined as

$$E(w) = \frac{1}{N}\sum_{i=1}^{N} \quad L(x_i, {x_i}^{+}, {x_i}^{-}; w) + \frac{\lambda}{2}||w||^2,$$

where the loss function is a modified hinge loss

$$L(x_i, {x_i}^{+}, {x_i}^{-}; w) = max(0, \gamma - S(x_i, {x_i}^{+}; w) + S(x_i, {x_i}^{-}; w)).$$

In the above objective function, we want to find the parameters w, such that the energy function $E(w)$ over the training dataset is minimized. This indicates that for each training examples $(x_i, {x_i}^{+}, {x_i}^{-})$, the similarity $S(x_i, {x_i}^{+}; w)$ tends to be higher than $S(x_i, {x_i}^{-}; w)$. In our practice, the parameters w are updated time after time using the method Stochastic Gradient Descent, where the following formulas are applied

$$\underline{w}_{t+1} \leftarrow \mu \underline{w}_t + \eta \frac{\partial E}{\partial w_t},$$

$$w_{t+1} \leftarrow w_t - \underline{w}_t.$$

The notations λ, γ, μ and η are the hyper parameters selected according to the empirical experiments.

Experiment

Decision Tree

Dataset

To validate the idea of building explicit rules via learning decision tree, we test it on the Iris Plants database [18]. It contains 150 data instances of 3 classes of Iris flowers. As the dataset does not contain too many data instances or attributes, it is possible for us to manually design the rules for Iris classification, which is essential in implementing method based on classic logic. The following lists some examples about how an Iris may look like in each category.

An illustration of the flower Iris

Sepal		Petal		Category
Length (cm)	Width (cm)	Length (cm)	Width (cm)	
5.1	3.5	1.4	0.2	Iris Setosa
4.9	3.0	1.4	0.2	Iris Setosa
7.0	3.2	4.7	1.4	Iris Versicolor
6.4	3.2	4.5	1.5	Iris Versicolor
6.5	3.2	5.1	2.0	Iris Virginica
6.4	3.2	5.3	2.3	Iris Virginica

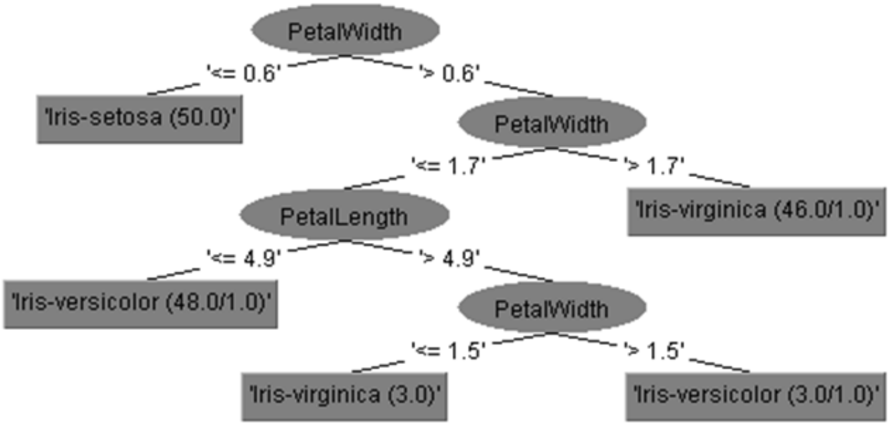
Some examples in the Iris Plants dataset

Evaluation

The classification error is used for evaluating the performance, which is the percentage of images that are incorrectly classified. A standard 3-fold cross validation is used to estimate how well the model can perform on unseen data. Namely, the database is even divided into 3 folds. Each time we choose the data in two folds for training and the rest for testing. The process is repeated for three times, and the average performance is reported as the final result.

Results

We compare the decision tree learning based method (C4.5) to classic logic based method, as well as traditional supervised learning methods introduced in the class. The following figure shows the decision tree learned from the Iris Plants dataset.



The decision tree learned on the Iris Plants dataset

The learned decision tree tells us we can somehow make good classifications by simply checking the characteristics of the petal of an Iris. Besides, the table below summarizes the results of different methods.

Method	Classification Error (%)	Modeling Time
Classic Logic	50.01	About 5 minutes
Decision Tree	12.34	0.01s
Random Forest (100 Trees)	8.22	0.13s
Bayes Network	8.10	0.01s
Naïve Bayes	9.37	<1ms
1-Nearest Neighbor	10.65	0s
3-Nearest Neighbor	8.24	0s

The classification results of different methods on the Iris Plants dataset

As shown in the table, since the manually designed rules may not be accurate enough given limited time for exploring the data, the Classic Logic based method does not perform so well. As for the decision tree based method, it produces significantly better result while less time is taken for building the model. We also test the method Random Forest, where 100 randomly generated decision trees are used to make the prediction based on the majority votes. The randomness exists as only a random subset of attributes are selected as candidates in each split. It is observed that the performance further improves with a slight loss of time efficiency. In addition, we test the traditional methods introduced in the class, such as Bayes Network, Naïve Bayes and k-Nearest Neighbors. We find that the Bayes Network performs the best among these methods. But in terms of interpretability of decision making, the decision tree based method is still the best option.

Deep Neural Network

Dataset

To evaluate the performance of similarity learning via Deep Neural Network, three databases are used in the experiments. Namely, the MNIST [20] dataset, the CIFAR-10 [21] dataset and the PASCAL VOC 2007 [22] dataset. The MNIST dataset contains 10 classes of handwritten digits, including 60,000 images for training and 10, 000 images for testing. The CIFAR-10 dataset are images of 10 classes of objects. The classes are completely mutually exclusive in the sense that they do not have hierarchical relationships. Also, there are 60,000 training images and 10,000 testing images. The PASCAL dataset consists of 20 classes of objects. It is more challenging as one image may contain multiple kinds of objects. There are 5,000 images for training and 5,000 images for testing in this dataset.

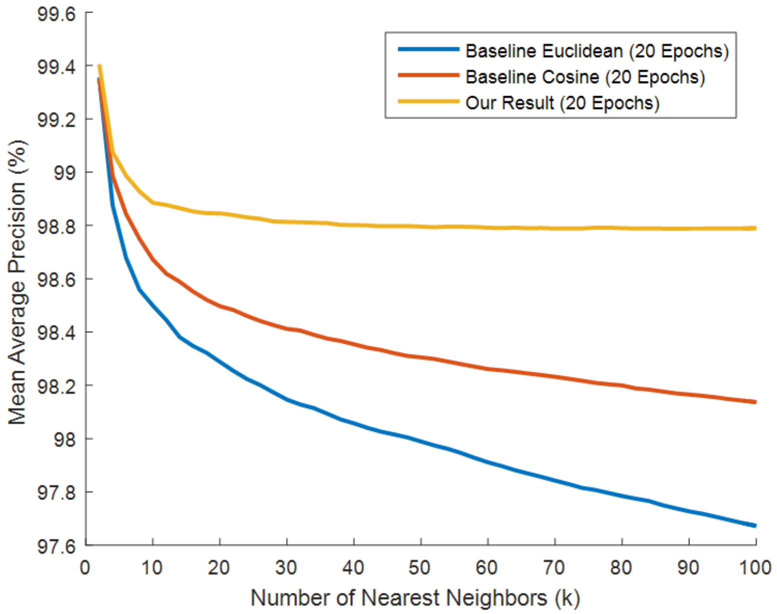
Evaluation

The performance is evaluated in terms of image retrieval accuracy, which is measured by the Mean Average Precision (mAP) in the k-Nearest Neighbors (k-NN). Namely, for each query, we can find its k-Nearest Neighbors using the learned similarity measure. And we can compute the percentage (Precision) of the relevant images among the retrieved results. The average retrieval precision of all queries is denoted as the mAP.

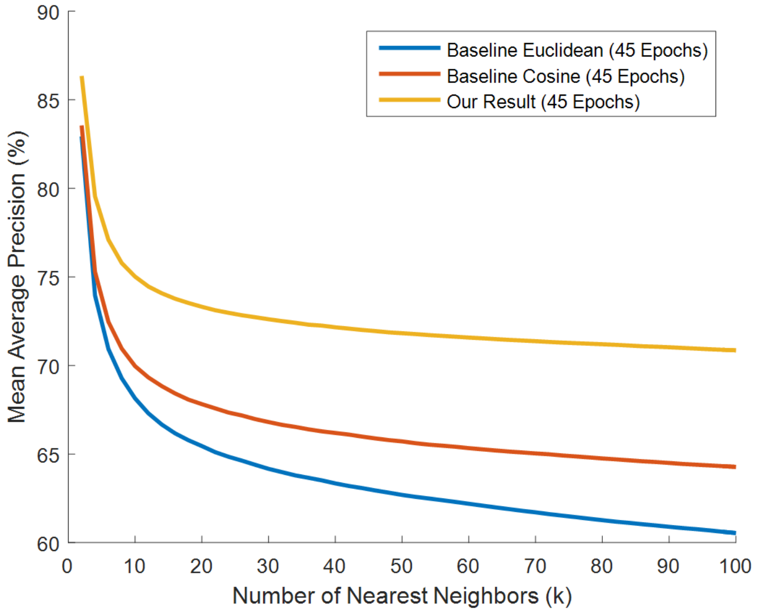
Results

Before learning a CNN, we have to first decide the structure of the network. For the MNIST and CIFAR-10 dataset, the network structure of LeNet is utilized while for the PASCAL dataset, the network structure of VGG-f is borrowed. The baseline we used for comparison is the CNN trained by traditional method aimed at minimizing the classification error. The output of the last layer is used as the vector presentation of each image. The Euclidean distance and the Cosine distance are used to find the k-Nearest Neighbors of a query. For fair comparison, we use the same training and test data with the same network structure, and they are trained the same number of times (epochs). The only difference is that our goal is to minimize the overall loss in terms of similarity instead of classification error.

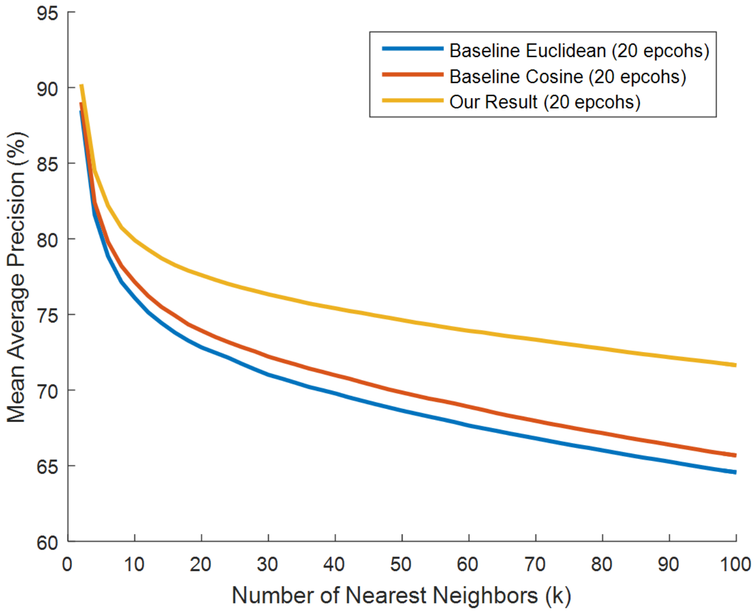
The following figures show the results on the three datasets respectively. For each dataset, the mAP is plotted as a function of the number of nearest neighbors k.



The results on the MNIST dataset

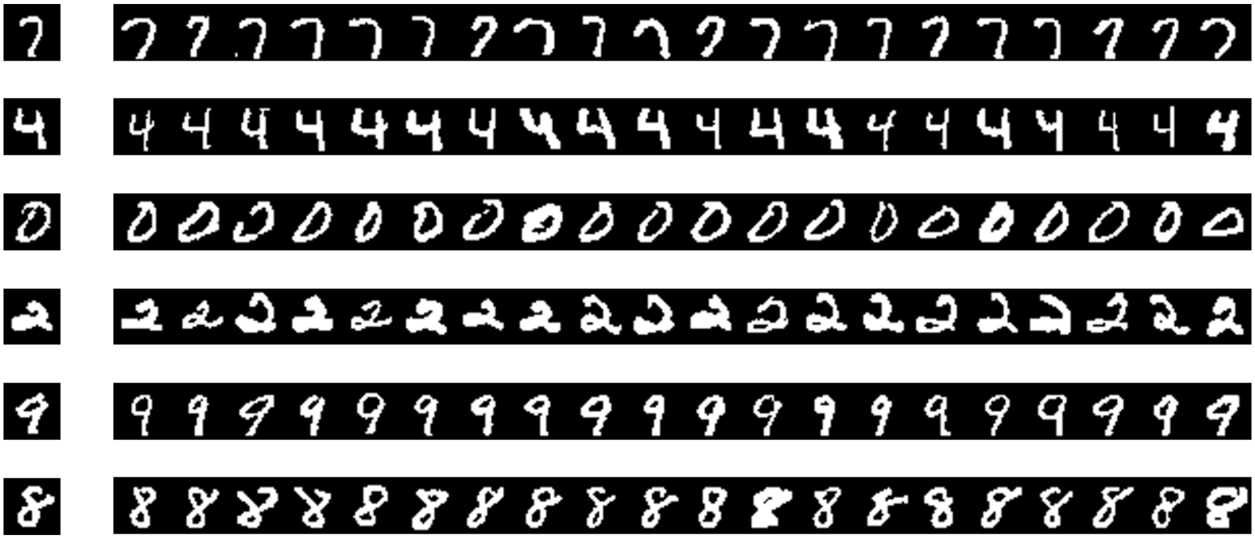


The results on the CIFAR-10 dataset

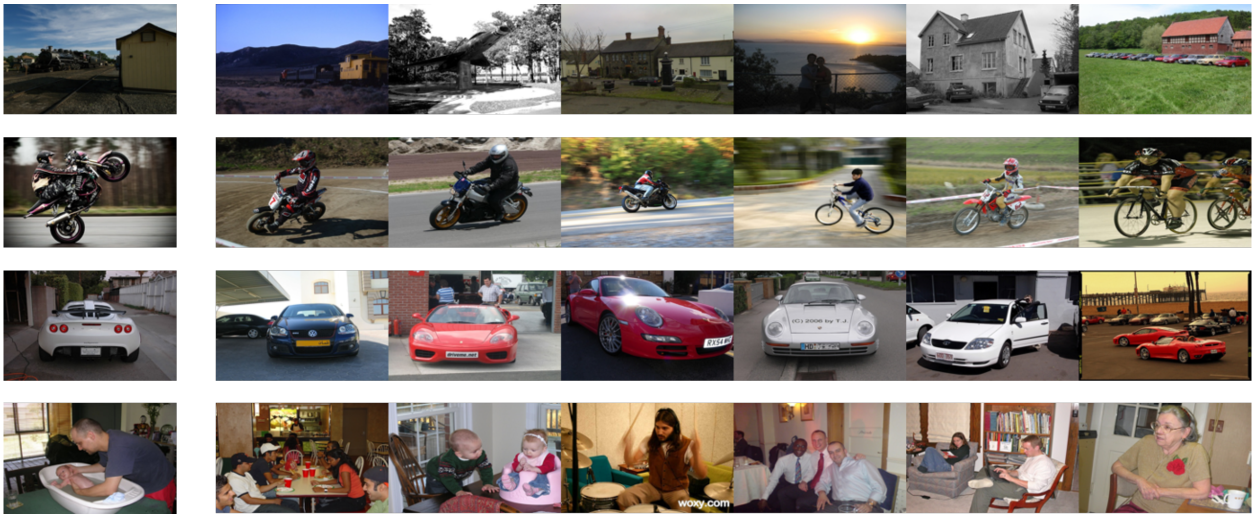


The results on the PASCAL dataset

It is observed that our method significantly improves the results compared to the baseline. This validates the effectiveness of the applied similarity learning method. In addition, we present some qualitative results for better understanding the performance, as shown in the following figures.



Exemplar results of image retrieval on the MNIST dataset



Exemplar results of image retrieval on the PASCAL dataset

In the above examples, images in the left column are the query images while the images one the right are the k-most similar images found by our method. For the handwritten images, the learned CNN achieves almost perfect results. And for the images in the PASCAL dataset, which may be more common to see in our daily life, it also achieves reasonable results. Although there are some mistakes such as treating bikes as motor bikes, but these mistakes could be explained and they are not counter-intuitive.

Discussions

In this section, we discuss the advantages and disadvantages of the applied methods, as a summary of the lessons learned in this project.

Decision Tree

In a constructed decision tree, we can clearly see how decisions are made in each step of internal inference process. Also, it is not necessary to provide a large amount of training data in order to obtain a reasonable performance.

On the other hand, the overfitting problem may occur if we do not limit the size of the induced tree by certain pruning methods. And it is hard to determine the proper size of the decision tree or the parameters in pruning the tree in practice. Also, the number of instances in each class must be balanced. Otherwise it may favour the instances in the majority class and make biased predictions.

Deep Neural Network

Deep neural network is a powerful tool for modeling many AI problems and they are widely used in recent years. One reason is that it can achieve superior performance compared to traditional methods by just feeding enough data and training enough times. And in our application, we do not need to manually design the image features or similarity metric in the task of image retrieval.

But there are also potential issues in learning a deep neural network. First, it may be computationally expensive to optimize the parameters in a DNN, as the number of them could be huge. For example, there are 4.3×10^5 parameters in LeNet, 6.1×10^7 parameters in VGG. Often we may need additional computational resources like GPU or Cloud computing for solving problems in real-time. Secondly, we may need a lot more training data than traditional methods in order to optimize these parameters, according to the theory in statistics. Also, the structure of the neural network has to be carefully engineered. And currently there are no theoretical guidance for the design, thus it looks more like an art than a science. Besides, there are a few more parameters to be selected with care so as to achieve reasonable performance, such as the learning rate, the momentum, the weight decay, the regularization parameter, the number of training epochs, etc. If these parameters are chosen improperly, the problem of underfitting and overfitting may also occur.

Conclusion

In this project, we investigated several learning based reasoning methods. First, we explore the way to build explicit rules for problem solving. It corresponds to the learning process of conscious reasoning and the decision tree based technique is applied in our project. As conscious reasoning may not solve certain problems in our practice, we also investigate the way to build implicit rules for problem solving. More specifically, we utilize the deep neural network to simulate the process of unconscious reasoning. In addition to problem solving, emotion and creativity are also important measures of human intelligence. Thus, we conduct a short survey about some recent progress in the related research community.

References

[1] McCulloch, Warren S., and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." The bulletin of mathematical biophysics 5.4 (1943): 115-133.

[2] Rumelhart, David E., et al. "Sequential thought processes in PDP models." Vol. 2 (1986): 3-57.

[3] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[4] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.

[5] Jozefowicz, Rafal, et al. "Exploring the limits of language modeling." arXiv preprint arXiv:1602.02410 (2016).

[6] Dahl, George E., Navdeep Jaitly, and Ruslan Salakhutdinov. "Multi-task neural networks for QSAR predictions." arXiv preprint arXiv:1406.1231 (2014).

[7] "Toxicology in the 21st century Data Challenge"_, National Center for Advancing Translational Science. Web. 25 April, 2016.

[8] D. Ciresan, U. Meier, J. Schmidhuber., "Multi-column Deep Neural Networks for Image Classification," Technical Report No. IDSIA-04-12, 2012.

[9] Damasio AR. Descartes' Error: Emotion, Reason, and the Human Brain. New York: Grosset & Putnam, (2014).

[10] Knutson, Brian, et al. "Neural predictors of purchases." Neuron 53.1 (2007): 147-156.

[11] Greene, Joshua D., et al. "The neural bases of cognitive conflict and control in moral judgment." Neuron 44.2 (2004): 389-400.

[12] Boden, Margaret A. "Creativity and artificial intelligence." Artificial Intelligence 103.1 (1998): 347-356.

[13] Kurzweil, Ray. The age of intelligent machines. Vol. 579. Cambridge: MIT press, 1990.

[14] Hinton, G. et al. Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine 29, 82–97 (2012).

[15] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[16] Tom Geller. “How Do You Feel? Your Computer Knows”. Communications of the ACM, Vol. 57 No. 1, Pages 24-26, 2014.

[17] Littlewort, G., Whitehill, J., Wu, T., Fasel, I., Frank, M., Movellan, J., and Bartlett, M. “The computer expression recognition toolbox (CERT)”. In Proceedings of the 9th IEEE Conference on Automatic Face & Gesture Recognition, pp. 298-305 (Santa Barbara, California, 2011)

[18] P. Ekman and W. Friesen. The Facial Action Coding System: A Technique For The Measurement of Facial Movement. Consulting Psychologists Press, Inc., San Francisco, CA, 1978.

[19] Fisher, R. A., and Michael Marshall. "Iris plants database." Donated to the UCI Machine Learning Database Repository University of California, Irnive (1995) by Michael Marshall (1988).

[20] LeCun, Yann, Corinna Cortes, and Christopher JC Burges. "The MNIST database of handwritten digits." (1998).

[21] Krizhevsky, Alex, and G. Hinton. "Convolutional deep belief networks on cifar-10." Unpublished manuscript 40 (2010).

[22] Everingham, M., et al. "The pascal visual object classes challenge 2007 (voc 2007) results (2007)." (2008).

[23] Loffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167 (2015).

[24] Quinlan, J. Ross. C4. 5: programs for machine learning. Elsevier, 2014.

[25] Quinlan, J. Ross. "Induction of decision trees." Machine learning 1.1 (1986): 81-106.

[26] Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2.3 (2002): 18-22.

[27] Ma, Junshui, et al. "Deep neural nets as a method for quantitative structure–activity relationships." Journal of chemical information and modeling 55.2 (2015): 263-274.

[28] Xiong, Hui Y., et al. "The human splicing code reveals new insights into the genetic determinants of disease." Science 347.6218 (2015): 1254806.

[29] Lenat, Douglas B., and John Seely Brown. "Why AM and EURISKO appear to work." Artificial intelligence 23.3 (1984): 269-294.

[30] Simon, Herbert Alexander. "Discovery, invention, and development: Human creative thinking." Proceedings of the National Academy of Sciences 80.14 (1983): 4569-4571.

[31] Daniel Miller. “Silicon symphony: Music composed by computer monitoring people's emotional response to Beethoven to be played in public for the first time tonight.” Daily Mail. Daily Mail, 23 February 2013.

[32] Salevati, Sara, and Steve DiPaola. "A creative artificial intelligence system to investigate user experience, affect, emotion and creativity." Proceedings of the Conference on Electronic Visualisation and the Arts. British Computer Society, 2015.

[33] Tresset, Patrick, and Frederic Fol Leymarie. "Portrait drawing by Paul the robot." Computers & Graphics 37.5 (2013): 348-363.