Abstract

This article demonstrates a new conceptor network based classifier in classifying images. Mathematical descriptions and analysis are presented. Various tests are experimented in this context using three benchmark datasets: MNIST, CIFAR-10 and CIFAR-100. The experiments displayed that conceptor network can offer superior results and flexible configurations than conventional classifiers such as Softmax Regression and Support Vector Machine (SVM).

1 Introduction

Recent successes in deep learning has revolutionized the field of machine learning and pattern recognition\cite{1,2,3,4,5,6,7}. In most successful classification applications, softmax regression\cite{8} and support vector machine (SVM)\cite{9} are usually employed and achieved state-of-the-art performance\cite{2,3,7}.

One major disadvantage of current popular classifiers (i.e. softmax regression) is that they are not extensible once the learning architecture is built. In a supervised learning task, current classifiers are not capable of modeling new class of data by default. They have to reinitialize the entire architecture or pay expensive cost on retraining the learning model. Another disadvantage of current classifiers is that they fail to model logical relationships in between classes. This usually prevents us from conducting semantic reasoning directly using non-semantic information (i.e. image feature, audio feature).

This article utilized a naive conceptor network\cite{10} as classifier for image classification benchmarks (details in Section\ref{sec:2}). The network generates a conceptor for each presented class. As the name suggested, a conceptor represents the “concept” of a class. And then learned conceptors are used to measure relevance between data and the classes. Three popular datasets: MNIST\cite{11}, CIFAR-10\cite{12} and CIFAR-100\cite{12} are experimented and analyzed (Section\ref{sec:3}). The experiments showed that conceptor network is capable of achieving comparable performance on MNIST and superior performance on CIFAR-10 and CIFAR-100 datasets using simple features from shallow neural networks (Section\ref{sec:3.1}-\ref{sec:3.3}). The model is capable of learning new unseen classes without damaging the performance of previous learned classes (Section\ref{sec:3.4}). In the addition experiment of CIFAR-100, the learned sub-class conceptors can classify super-class label without retraining by combining them using logical operation (Section\ref{sec:3.5}).
2 Method

2.1 Naive conceptor network

Consider a modified recurrent network:

\[ h_t = \sigma(W_{hh} h_{t-1} + W_{ih} x_t + b) \]  
\[ y_t = h_t \]

where \( \sigma(\cdot) \) is the activation function (\( \tanh \) in this context). \( x_t \in \mathbb{R}^n \) is an \( n \)-dimensional input at time \( t \), \( h_t \in \mathbb{R}^m \) is an \( m \)-dimensional hidden activation input at \( t \), the output of the network \( y_t \) is a replica of hidden activation. \( W_{hh} \) and \( W_{ih} \) are weight matrices from hidden to hidden and from input to hidden correspondingly. Note that all weight matrices are initialized randomly and not subjected to change over time.

Let \( X = \{ x_t \}_{t=1}^{T} \) is a time-dependent pattern or a class of data (i.e. object, face, scene, etc) and \( H = [h_t]_{t=1}^{T} \in \mathbb{R}^{m \times T} \) (\( h_0 \) is randomly initialized) as respective hidden activation history. The hidden activation can be modeled by following cost function:

\[ J(C, \alpha, X) = \frac{1}{T} \sum_{t=1}^{T} ||h_t - Ch_t||^2 + \lambda ||C||^2_\alpha \]

where \( C \) is called conceptor that characterize data \( X \), \( \lambda \in (0, 1) \) is the regularization term.

Given Eqn. 3, conceptor \( C \) can be computed by employing stochastic gradient descent (SGD) by

\[ C' = C + \alpha \frac{\partial}{\partial C} J \]

where \( \alpha \) is learning rate.

Since Eqn. 5 presents a convex problem, there is an analytical solution given by [10][13]:

\[ C = R(R + \lambda I)^{-1} \]

where \( R \in \mathbb{R}^{m \times m} \) is correlation matrix that is computed by

\[ R = \frac{HH^T}{T} \]

Eqn. 4 and Eqn. 5 are expected to give the same solution under regularization. Considering the speed issue and size of datasets that are used in this paper, the analytical solution is primarily used.

2.2 Use logical operations over conceptor

A big advantage that is offered by conceptor network is enabling boolean operations [10]. Basic boolean operations such as NOT (\( \neg \)), AND (\( \land \)), OR (\( \lor \)) are usable over learned conceptors.

Given conceptors \( B \), \( C \) and identity matrix \( I \), operations that are suggested in [10] are listed here:

- **NOT**
  \[ \neg C = I - C \]

- **AND**
  \[ C \land B = (C^{-1} + B^{-1} - I)^{-1} \]

- **OR**
  \[ C \lor B = -(\neg C \land \neg B) \]
2.3 Use conceptor network as supervised classifier

Conceptor network takes a regression analysis on hidden activation of a class of data. The resulted conceptor has the property of quantifying relevance between a hidden activation using vector projection. This advantage can be taken to formulate a supervised classifier.

Let $\mathcal{C} = \{C_j\}_{j=1}^N$ as a collection of conceptors. Each conceptor $C_j$ characterizes one class of data. Let $\mathcal{D} = \{D_j\}_{j=1}^N$ as a collection of negative conceptors that produced by $C$:

$$D_j = \neg (C_1 \lor C_2 \lor \ldots \lor C_{j-1} \lor C_{j+1} \lor \ldots \lor C_N)$$

(10)

where $D_j$ represents the condition of not being other $N - 1$ classes.

Given a conceptor $C_j$, a negative conceptor $D_j$ and data $x$’s hidden activation $h$, a positive evidence is computed by

$$E^+(C, h) = h^\top C h$$

(11)

Correspondingly, negative evidence is

$$E^-(D, h) = h^\top D h$$

(12)

And combined evidence is computed by

$$E(C, D, h) = E^+(C, h) + E^-(D, h)$$

(13)

Positive evidence quantify the degree of relevance between signal $x$ and conceptor $C$. Negative evidence quantify the degree of relevance between signal $x$ and not being other classes than $C$.

Given data $x$’s hidden activation $h$, the signal belongs to class $j^*$:

$$j^* = \arg \max_j E(C_j, D_j, h) \quad j = 1, \ldots, N$$

(14)

where $C_j \in \mathcal{C}$ and $D_j \in \mathcal{D}$.

3 Experiments

In this section, experiments on MNIST, CIFAR-10 and CIFAR-100 are presented and discussed. All images are rescaled in $[0, 1]$. Images in CIFAR-10 and CIFAR-100 are converted in gray-scale. No further pre-processing is performed.

Image features are extracted from either a one-hidden-layer auto-encoder/a Convolutional Neural Network (ConvNet). ConvNet in this article has two convolution layers, one fully connected layer and one softmax layer, features are extracted from the fully connected layer. ConvNet is employed here to produce better image features than auto-encoder. The naming method of feature is [AE/ConvNet]-Feature-[Dimension of each feature] (e.g. AE-Feature-200 represents a set of auto-encoder features with 200 dimensions). Conceptor network directly takes features of images for classification.

There are 3 types of experiments are performed for each dataset:

1. Classification performance V.S. Number of neurons in reservoir
2. Classification performance V.S. Input feature size
3. Classification performance V.S. Feature extraction method (AE feature/ConvNet feature)

And 2 additional experiments are performed in order to test logical operations:

1. Learning new class without damaging previous learned classes (CIFAR-100 super-classes).
2. Use sub-class conceptors represent super-class conceptors (CIFAR-100)
3.1 Classify MNIST

MNIST dataset contains 60,000 training samples and 10,000 testing samples [11]. Each sample is a hand-written digit from 0-9. Experiments in this section used first 50,000 samples of training dataset and full testing dataset. The conceptor network generated 10 conceptors for total 10 classes given a set of features. Results are presented in Figure 1 and Table 1.

In Fig. 1 performance of conceptor network is stable across a large range of number of neurons in reservoir. And the network works well when the number of neurons is small. Results in Tab. 1 shows that Auto-encoder feature is slightly better than ConvNet feature. The performance here is not as good as some deep architectures [11, 14]. It offered similar performance to networks that have similar depth (3 layer MLP networks used in [11]).

![Figure 1: MNIST Classification Results](image)

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Number of hidden neurons</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE-Feature-200</td>
<td>125</td>
<td>96.85%</td>
</tr>
<tr>
<td>ConvNet-Feature-200</td>
<td>20</td>
<td>95.77%</td>
</tr>
<tr>
<td>AE-Feature-500</td>
<td>10</td>
<td><strong>97.49%</strong></td>
</tr>
<tr>
<td>ConvNet-Feature-500</td>
<td>10</td>
<td>95.93%</td>
</tr>
</tbody>
</table>

3.2 Classify CIFAR-10

CIFAR-10 consists of 60,000 $32 \times 32$ color images in 10 classes [12]. There are 50,000 training images and 10,000 testing images. CIFAR-10 is among the most popular benchmarks in pattern recognition. Like MNIST, experiments here also employ features from shallow networks. And results are showed in Fig. 2 and Table 2.

ConvNet features are performing better than Auto-encoder features in these experiments (Tab. 2). The best result is produced by ConvNet-Feature-500 with accuracy of **98.39%**. Classification results presented here have reached or been better than state-of-the-art methods [15, 16, 17, 18] (Tab. 2).
3.3 Classify CIFAR-100

CIFAR-100 consists of 60,000 $32 \times 32$ color images in 100 classes [12]. Each class has 500 training images and 100 testing images. And the 100 classes are grouped into 20 super-classes. There are many previous attempts on improving classification accuracy for CIFAR-100 [15, 16, 18, 19]. The results are presented in Figure 3 and Table 3. Due to some memory issue, CIFAR100 sub-classes classification is not carried out fully.

Similar to previous results in CIFAR-10, ConvNet features in this section work a lot better (7%-10%) than Auto-encoder features. Both best results of super-class and sub-class classification are generated by ConvNet-Feature-500 (96.48% and 83.04% respectively). Performance of proposed method is better than previous published results (Tab. 4).

### Table 2: CIFAR-10 Results

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Number of hidden neurons</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE-Feature-200</td>
<td>5</td>
<td>85.04%</td>
</tr>
<tr>
<td>ConvNet-Feature-200</td>
<td>5</td>
<td>98.30%</td>
</tr>
<tr>
<td>AE-Feature-500</td>
<td>490</td>
<td>88.09%</td>
</tr>
<tr>
<td>ConvNet-Feature-500</td>
<td>100</td>
<td>98.39%</td>
</tr>
</tbody>
</table>

### Table 3: CIFAR-100 Super-classes Results

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Number of hidden neurons</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE-Feature-200</td>
<td>365</td>
<td>87.67%</td>
</tr>
<tr>
<td>ConvNet-Feature-200</td>
<td>5</td>
<td>94.81%</td>
</tr>
<tr>
<td>AE-Feature-500</td>
<td>10</td>
<td>88.99%</td>
</tr>
<tr>
<td>ConvNet-Feature-500</td>
<td>20</td>
<td>96.48%</td>
</tr>
</tbody>
</table>
Table 4: CIFAR-100 Sub-classes results

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Number of hidden neurons</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE-Feature-200</td>
<td>10</td>
<td>65.28%</td>
</tr>
<tr>
<td>ConvNet-Feature-200</td>
<td>10</td>
<td>76.59%</td>
</tr>
<tr>
<td>AE-Feature-500</td>
<td>25</td>
<td>62.96%</td>
</tr>
<tr>
<td>ConvNet-Feature-500</td>
<td>20</td>
<td>83.04%</td>
</tr>
</tbody>
</table>

3.4 Learn new classes incrementally

Another advantage of conceptor network is that it can learn new classes without damaging previous learned classes. With this feature, a learning system can be easily scaled up without taking many efforts. This experiment takes 100 sub-classes of CIFAR-100 to demonstrate incremental learning.
The network is initially trained with 2 of all 100 classes. Then a conceptor is computed when the network observed a new class. The new conceptor is then appended in the list of existed conceptors. And it can be used to classify new data along with old ones. Fig. 5 presents the result of the experiment. The network in this experiment consists of 180 hidden neurons. The blue line is the result generated by ConvNet-Feature-200 (average accuracy: 77.55%) and the red line is the result generated by ConvNet-Feature-500 (average accuracy: 83.81%). Both results are stable during learning new classes incrementally.

3.5 Model super-classes directly from sub-classes in CIFAR-100

One feature of CIFAR-100 is relationship between super-class and sub-class. As discussed previously, each super-class corresponds to 5 sub-classes. Conventionally, a mapping table is used when a classifier generates (e.g. softmax) super-class label from sub-class label of an image.

In this experiment, logical operation OR is used to model super-classes directly from sub-classes. Here the conceptors established a semantic relationship between super and sub-classes instead of a simple mapping method. One can use this property of conceptor to model complex relationships between data categories. The result is presented in Fig. 6. The classification performance stays high and stable across large number range of hidden neurons.
4 Conclusion

This article demonstrated our first attempt of using conceptor network in pattern recognition benchmarks. Conceptor network exhibited impressive performance on classification of tiny object images. The proposed classifier showed superior performance with shallow network features. It can learn new class of data incrementally without harming previous learned classes. And it is capable of modeling semantic relationships by using logical operations over learned conceptors.

The potential of conceptor network is unlimited by current presentation. Our future works include exploring theory and applications of conceptor network. Many state-of-the-art supervised application now potentially can replace conventional classifiers with conceptor network.
References


