Artificial Superintelligence: Network of Intelligent Computers That Self-Improve

Poondru Prithvinath Reddy

October 3, 2020

Artificial Superintelligence: Network of Intelligent Computers That Self-Improve

Poondru Prithvinath Reddy

ABSTRACT

Artificial Super Intelligence or ASI that is more potent and refined than human's intelligence. ASI is based on the ideas that machines can imitate the human mind, their way of working to the extent that they can even supersede them. Today networking architectures is becoming smart and intelligent with the help of Internet Of Things (IoT) and Artificial Intelligence(AI), one such applications is the integration of IoT and AI in a Smart City project where various cameras and sensors installed at different spots and connect them to data center servers that can make intelligent decisions based on the inputs from the cameras and sensors. In such structure, the IoT devices handle basic recognition and next level sophisticated inputs are sent to remote servers. In this paper, we implement object detection CNN-SVM with different SVM architectures running on different computers on a network and the camera (IoT) provide a stand-alone processing power for street object detection which is the core function of traffic systems and public safety in a city. A Convolutional Neural Network model works on sensory device and SVMs work on remote servers. The test results show good accuracy. Since sensory devices work round the clock in a city environment, there is no practical limit on how much CNN-SVM can learn and greater integration of AI into network architectures can help develop into cognitive networks that will show network-wide intelligent behavior with an ability for self-improvement.

INTRODUCTION

"**Superintelligence**" refers to the idea that steady advances in artificial intelligence, or machine (computer) intelligence, might one day result in creating a machine vastly superior to humans in reasoning and decision-making abilities.

Artificial Super Intelligence or ASI that has the capability to perform the tasks that are impossible for the human mind to think or do. It is that aspect of intelligence that is more potent and refined than a human's intelligence. Superintelligence is capable of outperforming human intelligence; it is extremely powerful in doing that. The human brain is made of neurons and is limited to some billion neurons. Superintelligence, therefore challenges this trait, which knows no limit.

The road to endless possibilities of Artificial Super Intelligence is paved by the ideas that machines can imitate the human mind, their way of working to the extent that shortly they can even supersede them. Under these circumstances, it is inevitable that ASI will be much better in concluding tasks that humankind would fail to achieve, and will function in better ways compared to the human. In its first step, Artificial Super Intelligence aims to improve the cognitive abilities of the machine. In the future, the ASI will become more conscious, self-sustainable, and self-learning, developing, and improving constantly.

Edge-based street object detection

Nowadays everything is becoming smart and intelligent with the help of Internet of Things (IoT) and artificial intelligence (AI). One of the promising applications of the integration of IoT and AI is smart city. The typical design pattern of smart cities is to install cameras and various sensors as many spots as possible and connect them to data center servers that can make smart decisions based on the inputs from the cameras and sensors. In such structure, network bandwidth may hinder the real-time processing because sensory data should be sent to the servers in the remote location before making any decision. To solve this problem, recent a few studies demonstrated edge computing. In edge computing, IoT devices can handle basic recognition. Thus, only sophisticated inputs that are not handled by IoT devices are sent to remote servers. In addition, the edge computing can provide stand-alone processing power for handling one of the fundamental applications of smart city, which is street object detection. Correct detection of various street objects is the core function of traffic systems and public safety applications of smart cities and a convolutional neural network model is developed by training with such data.

Artificial Intelligence Enabled Networking

With today's computer networks becoming increasingly dynamic, heterogeneous, and complex, there is great interest in deploying artificial intelligence (AI) based techniques for optimization and management of computer networks. AI techniques—that subsume multidisciplinary techniques from machine learning, optimization theory, game theory, control theory, and meta-heuristics—have long been applied to optimize computer networks in many diverse settings. Such an approach is gaining increased traction with the emergence of novel networking paradigms that promise to simplify network management (e.g., cloud computing, network functions virtualization, and software-defined networking) and provide intelligent services (e.g., future 5G mobile networks). Looking ahead, greater integration of AI into networking architectures can help develop a future vision of cognitive networks that will show network-wide intelligent behavior to solve problems of network heterogeneity, performance, and quality of service (QoS).

Object Recognition

Object recognition is a general term to describe a collection of related computer vision tasks that involve identifying objects in digital photographs.

Image classification involves predicting the class of one object in an image. *Object localization* refers to identifying the location of one or more objects in an image and drawing abounding box around their extent. *Object detection* combines these two tasks and localizes and classifies one or more objects in an image.

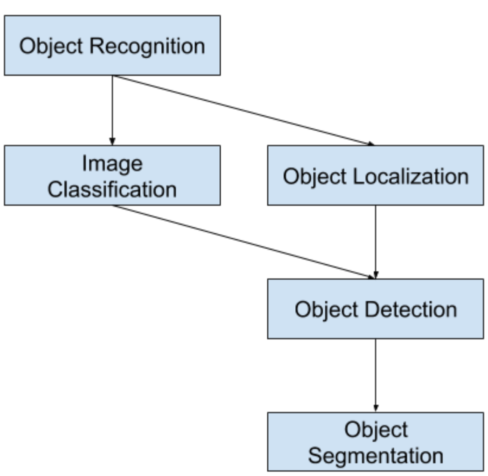
When a user or practitioner refers to “*object recognition*“, they often mean “*object detection*“.

As such, we can distinguish between these three computer vision tasks:

- **Image Classification**: Predict the type or class of an object in an image.
  - *Input*: An image with a single object, such as a photograph.
  - *Output*: A class label (e.g. one or more integers that are mapped to class labels).
- **Object Localization**: Locate the presence of objects in an image and indicate their location with a bounding box.
  - *Input*: An image with one or more objects, such as a photograph.
  - *Output*: One or more bounding boxes (e.g. defined by a point, width, and height).
- **Object Detection**: Locate the presence of objects with a bounding box and types or classes of the located objects in an image.
  - *Input*: An image with one or more objects, such as a photograph.
  - *Output*: One or more bounding boxes (e.g. defined by a point, width, and height), and a class label for each bounding box.

One further extension to this breakdown of computer vision tasks is *object segmentation*, also called “object instance segmentation” or “semantic segmentation,” where instances of recognized objects are indicated by highlighting the specific pixels of the object instead of a coarse bounding box.

From this breakdown, we can see that object recognition refers to a suite of challenging computer vision tasks.



Overview of Object Recognition Computer Vision Tasks.

Support Vector Machine (SVM)

Linear SVM is the newest extremely fast machine learning (data mining) algorithm for solving multiclass classification problems from ultra large data sets that implements an original proprietary version of a cutting plane algorithm for designing a **linear** support vector machine.

In machine learning, **Support Vector Machine** (**SVM**) is a **non-**probabilistic, **linear**, binary classifier used for classifying data by learning a hyperplane separating the data. Classifying a **non-linearly** separable dataset using a **SVM** – a linear classifier: However, it can be used for classifying a non-linear dataset also.

Linear classifier (SVM) is used when number of features are very high, e.g., document **classification**. This is because **Linear** SVM gives almost similar accuracy as **non linear** SVM but **Linear** SVM is very very fast in such cases and **non-linear classifier** is useful when data is not linearly separable.

METHODOLOGY

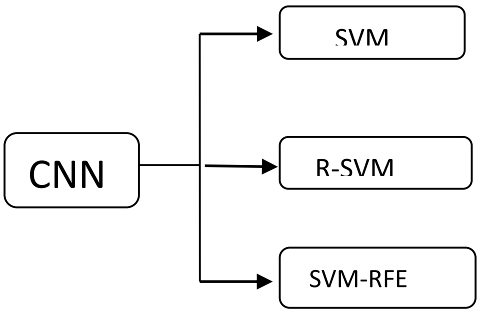
Meanwhile, the difference between an AGI and an ASI is that the latter, theoretically, will never exist in a physical form. It will operate entirely within a supercomputer or network of supercomputers. Depending on the goals of its creators, it may also get full access to all data stored on the Internet, as well as whatever device or human that feeds data into and over the Internet. This means there will be no practical limit to how much this ASI can learn and how much it can self-improve.

SVM

What is **SVM-RFE**?
The main purpose of **SVM-RFE (Support Vector Machine – Recursive Feature Elimination)** is to compute the ranking weights for all features and sort the features according to weight vectors as the classification basis.

Recursive feature elimination (RFE) is a **feature** selection method that fits a model and removes the weakest **feature** (or **features**) until the specified number of **features** is reached. RFE requires a specified number of **features** to keep, however it is often not known in advance how many **features** are valid. **SVM-RFE** is an iteration process of the backward removal of features.

We define following structure of CNN-SVM model for applying network-wide intelligence with different SVM implementations to achieve optimal method of object localization and recognition.



Support Vector Machine (SVM)

The support vector machine (SVM) was developed for binary classification. Its objective is to find the optimal hyperplane f(w, x)=w·x + b to separate two classes in a given dataset, with features $x \in R^m$.

SVM learns the parameters w by solving an optimization problem (Eq. 1).

$$min \frac{1}{p} w^T w + C \sum_{i=1}^{p} \max(0, 1 - y'_i (w^T x_i + b)) \qquad (1)$$

Where $w^T w$ is the Manhattan norm (also known as L1 norm), C is the penalty parameter (may be an arbitrary value or a selected value using hyper-parameter tuning), y' is the actual label, and $w^T x + b$ is the predictor function. Eq. 1 is known as L1-SVM, with the standard hinge loss. Its differentiable counterpart, L2-SVM (Eq.2), provides more stable results.

$$min \frac{1}{p} \|w\|_2^2 + C \sum_{i=1}^{p} \max(0, 1 - y'_i (w^T x_i + b))^2 \qquad (2)$$

Where ǁwǁ2 is the Euclidean norm (also known as L2 norm), with the squared hinge loss.

The methodology essentially consists of implementation of CNN-SVM method with a scheme of feature selection and classification, and compare its performance with R-SVM (Recursive SVM) on simulation data. However, we have not implemented SVM-RFE (Support Vector Method Recursive Feature Elimination) method.

ARCHITECTURE

Recursive CNN-SVM Architecture for Image Classification

MNIST is an established standard hand written digit classification dataset that is widely used for benchmarking deep learning models. However, we have used the Fashion-MNIST dataset. The said dataset consists of images having the same distribution, the same number of classes, and the same color profile as MNIST. Also, it is having 10,000 training examples and 10,000 test cases.

The Deep Artificial Neural Network

We used two convolutional layers:

- The first layer will have 32-5 x 5 filters,
- The second layer will have 64-5 x 5 filters

In addition, there are two max-pooling layers each of size 2 x 2.

We used a RELU as our activation function which simply takes the output of max_pool and applies RELU.

Fully connected layer:

Just like any other layer, we declare weights and biases as random normal distributions. In fully connected layer, we take all the inputs, do the standard operation on it. The Fully Connected Layer has 1024 Hidden Neurons.

We added Dropout into the network to overcome the problem of overfitting to some extent and also to improve the training and validation accuracy.

The final layer is Output Layer with 10 Output Classes.

At the last layer of the CNN, instead of the conventional softmax function with the cross entropy function (for computing loss), the L2-SVM is implemented. That is, the output shall be translated to the following case $y \in \{-1, +1\}$, and the loss is computed by Eq. 2.The weight parameters are then learned using Adam.

Machine Learning methods for feature selection and classification have been playing active roles in analyzing high-throughput data. We used both normal linear SVM and recursive support vector machine (R-SVM) to select input features for classification. The proposed R-SVM algorithm will recursively classify the training samples with SVM and select features according to their weights in the SVM classifier.

TEST RESULTS

Image Classification is about classifying objects in an image and the test results show good accuracy between training and validation data. However the CNN algorithm needs a lot of regions to predict accurately and hence high computation time.

We compared two methods i.e. SVM and R-SVM, the R-SVM adopting recursive procedures to select features in SVM classifiers. Although the two methods (SVM and R-SVM) did not differ significantly in their validation performances, it appears that R-SVM is more robust and can recover more informative features. The proposed R-SVM method is suitable for analyzing high-throughput data and it outperforms SVM in the robustness and in the ability to recover informative features.

CONCLUSION

Artificial Super Intelligence (ASI) is based on the idea that machines not only imitate the human mind but can even supersede human's intelligence. In order to achieve this, ASI will have to be more intelligent by improving intelligent abilities of the artificial machines. In this paper, we have implemented an object detection CNN-SVM architecture where street objects are detected by cameras (IoT) with stand-alone processing power in a smart city application and sophisticated inputs are handled by AI on remote servers. The test results show that it is possible to build network of intelligent computers in a city environment with integration of AI into networking architectures that can develop into cognitive networks which will show network-wide intelligent behavior with an ability to self-improvement.

REFERENCES

1. https://github.com/
2. Abien Fred M. Agarap "An Architecture Combining Convolutional Neural Network(CNN) and Support Vector Machine (SVM) for ImageClassification". https://arxiv.org/pdf/1712.03541.pdf