

[DESCRIPTION]

[Title of Invention]

METHOD FOR TRAINING NEURON NETWORK AND ACTIVE
LEARNING SYSTEM

[Technical Field]

[0001]

This invention relates generally to a method for training a neural network, and more specifically to an active learning method for training artificial neural networks.

[Background Art]

[0002]

Artificial neural networks (NNs) are revolutionizing the field of computer vision. The top-ranking algorithms in various visual object recognition challenges, including ImageNet, Microsoft COCO, and Pascal VOC, are all based on NNs.

[Summary of Invention]

[Technical Problem]

[0003]

In the visual object recognition using the NNs, the large scale image datasets are used for training the NNs to obtain good performance. However, annotating large-scale image datasets is an expensive and tedious task, requiring people to spend a large number of hours analyzing image content in a dataset because the subset of important images in the unlabeled dataset are selected and labeled by the human annotations.

[0004]

Accordingly, there is need to achieve better performance with less annotation processes and, hence, less annotation budgets.

[Solution to Problem]

[0005]

Some embodiments of the invention are based on recognition that an active learning using an uncertainty measure of features of input signals and reconstruction of the signals from the features provides less annotation processes with improving the accuracy of classifications of signals.

[0006]

Accordingly, one embodiment discloses a method for training a neuron network using a processor in communication with a memory, and the method includes determining features of a signal using the neuron network; determining an uncertainty measure of the features for classifying the signal; reconstructing the signal from the features using a decoder neuron network to produce a reconstructed signal; comparing the reconstructed signal with the signal to produce a reconstruction error; combining the uncertainty measure with the reconstruction error to produce a rank of the signal for a necessity of a manual labeling; labeling the signal according to the rank to produce the labeled signal; and training the neuron network and the decoder neuron network using the labeled signal.

[0007]

Another embodiment discloses an active learning system that includes a human machine interface; a storage device including neural networks; a memory; a network interface controller connectable with a network being outside the system; an imaging interface connectable with an imaging device; and a processor configured to connect to the human machine interface, the storage device, the memory, the network interface controller and the imaging interface, wherein the processor executes instructions for classifying a signal using the neural networks stored in the storage device, wherein the neural networks perform steps of

determining features of the signal using the neuron network; determining an uncertainty measure of the features for classifying the signal; reconstructing the signal from the features using a decoder neuron network to produce a reconstructed signal; comparing the reconstructed signal with the signal to produce a reconstruction error; combining the uncertainty measure with the reconstruction error to produce a rank of the signal for a necessity of a manual labeling; labeling the signal according to the rank to produce the labeled signal; and training the neuron network and the decoder neuron network using the labeled signal.

[0008]

Accordingly, one embodiment discloses a non-transitory computer-readable medium storing software comprising instructions executable by one or more computers which, upon such execution, cause the one or more computers to perform operations. The operation includes determining features of a signal using the neuron network; determining an uncertainty measure of the features for classifying the signal; reconstructing the signal from the features using a decoder neuron network to produce a reconstructed signal; comparing the reconstructed signal with the signal to produce a reconstruction error; combining the uncertainty measure with the reconstruction error to produce a rank of the signal for a necessity of a manual labeling; labeling the signal according to the rank to produce the labeled signal; and training the neuron network and the decoder neuron network using the labeled signal.

[0009]

In some embodiments, the use of an artificial neural network that determines an uncertainty measure may reduce central processing unit (CPU) usage, power consumption, and/or network bandwidth usage, which is advantageous for improving the functioning of a computer.

[Brief Description of the Drawings]

[0010]

[Fig. 1A]

FIG. 1A is a block diagram of the data flow of an active learning system for training a neural network in accordance with some embodiments of the invention.

[Fig. 1B]

FIG. 1B is a flowchart of an active learning system for training a neural network.

[Fig. 1C]

FIG. 1C is a block diagram of process steps to be performed based on some embodiments of the invention.

[Fig. 1D]

FIG. 1D shows a block diagram indicating an active learning process and a convolutional neural network (CNN) training process in accordance with some embodiments of the invention.

[Fig. 1E]

FIG. 1E is a block diagram indicating key process steps performed in an active learning system in accordance with some embodiments of the invention.

[Fig. 2]

FIG. 2 is a block diagram of an active method for ranking the importance of unlabeled images.

[Fig. 3]

FIG. 3 is a block diagram of a neural network to calculate the uncertainty of input signal according to some embodiments of the invention.

[Fig. 4]

FIG. 4 is a block diagram of a method for ranking the importance of unlabeled images in an active learning system according to some embodiments of the invention.

[Fig. 5]

FIG. 5 is a block diagram of an active learning system for annotating the unlabeled images in accordance with some embodiments of the invention.

[Fig. 6]

FIG. 6 is an illustration for the labeling interface.

[Fig. 7]

FIG. 7 shows an example of an accuracy comparison of active learning methods on CNN.

[Description of Embodiments]

[0011]

In some embodiments according to the invention, an active learning system includes a human machine interface, a storage device including neural networks, a memory, a network interface controller connectable with a network being outside the system. The active learning system further includes an imaging interface connectable with an imaging device, a processor configured to connect to the human machine interface, the storage device, the memory, the network interface controller and the imaging interface, wherein the processor executes instructions for classifying an object in an image using the neural networks stored in the storage device, in which the neural networks perform steps of determining features of a signal using the neuron network, determining an uncertainty measure of the features for classifying the signal, reconstructing the signal from the features using a decoder neuron network to produce a reconstructed signal, comparing the reconstructed signal with the signal to produce a reconstruction error, combining the uncertainty measure with the reconstruction error to produce a rank of the signal for a necessity of a manual labeling, labeling the signal according to the rank to produce the labeled signal, and training the neuron network and the decoder neuron network using the labeled signal.

[0012]

FIG. 1A shows an active learning system 10 in accordance with some embodiments of the invention. An initial setting of the active learning system 10 includes a neural network 100 initialized with random parameters, an initial set of labeled training images 101, a trainer 102, a set of unlabeled images 103. In this case, the neural network 100 is a user defined neural network.

[0013]

The active learning system 10 attempts to efficiently query the unlabeled images for performing annotations through the basic workflow shown in FIG. 1A. Based on the neural network (NN) 100 with randomly initialized parameters, the trainer 102 updates network parameters by fitting the NN 100 to the initial labeled training dataset of images 101. As a result, a trained NN 301 with the updated network parameters is used to rank the importance of images in an unlabeled dataset 103. The unlabeled images 103 are sorted according to importance scores 104 obtained from a ranking result performed by the trained NN 301. The K most important images 105 are stored into a labeling storage in a memory (not shown in the figure) associated to a labeling interface 106. In response to data inputs made by an operator (or annotator), the labeling interface 106 generates annotated images 107 having the ground truth labels. These annotated images 107 are then added to the initial labeled training dataset 101 to form a new training dataset 108. The trainer 102 then retrain the network 301 by fitting the new training dataset of images 108 and obtains updated neural network parameters 401. This procedure is iterative. The updated neural network parameters 401 are used to rank the importance of the rest of the unlabeled images 103, and the K most important images 105 are sent to the labeling interface 106. Usually, this procedure is repeated several times until a predetermined preferred performance is achieved or the budget for annotations is empty.

[0014]

Further, in some embodiments of the invention, a method for training a neuron network uses a processor in communication with a memory, and the method includes steps of determining features of a signal using the neuron network, determining an uncertainty measure of the features for classifying the signal, reconstructing the signal from the features using a decoder neuron network to produce a reconstructed signal, comparing the reconstructed signal with the signal to produce a reconstruction error, combining the uncertainty measure with the reconstruction error to produce a rank of the signal for a necessity of a manual labeling, labeling the signal according to the rank to produce the labeled signal, and training the neuron network and the decoder neuron network using the labeled signal. In some cases, the labeling can include labeling the signal using the neuron network if the rank does not indicate the necessity of the manual labeling process, and further the labeling can include transmitting a labeling request to an annotation device if the rank indicates the necessity of the manual labeling process.

[0015]

Further, the determining features may be performed by using an encoder neural network. In this case, the encoder neural network can perform feature analysis of given signals. In some cases, the signal may be an electroencephalogram (EEG) or an electrocardiogram (ECG). The neural network can use biological signals instead of image signals. Accordingly, some embodiments of the invention can be applied to provide specific signals for assisting a diagnosis of medical doctors.

[0016]

FIG. 1B is a flowchart of an active learning system for training neural network.

[0017]

The active learning system 10 attempts to efficiently query the unlabeled images for the annotation through a process flow shown in the figure. The process flow includes the following stages:

[0018]

S1 – An initial labeled training dataset is provided and the neural network is trained by using the dataset.

[0019]

S2 – By using the trained NN obtained in step S1, each image in the unlabeled dataset is evaluated and a score would be assigned to each image.

[0020]

S3 – Given the score obtained in step S2, images with the top K highest scores are selected for labeling by the annotation device.

[0021]

S4 – The selected images with newly annotated labels are added into the current (latest) labeled training set to get a new training dataset.

[0022]

S5 – The network is refined or retrained based on the new training dataset.

[0023]

As shown in FIG. 1B, the active learning algorithms of the active learning system 10 attempt to efficiently query images for labeling images. An initialization model is trained on an initial for small labeled training set. Based on the current model, which is the latest trained model, the active learning system 10 tries to find the most informative unlabeled images to be annotated. A subset of the informative images are labeled and added to the training set for the next round of training. This training process is iteratively performed, and the active learning system 10 carefully adds more labeled images for gradually increasing the accuracy performance of the model on the test dataset. By the very nature, the algorithms of

the active learning system 10 usually work much better than the standard approach for training, because the standard approach simply selects the samples at random for labeling.

[0024]

Although a term “image” is used in the specification, another “signal” can be used in the active learning system 10. For instance, the active learning system may process other signals, such as an electroencephalogram (EEG) or an electrocardiogram (ECG). Instead of the images, the EEG or ECG signals can be trained in the active learning system 10. Then the trained active learning system 10 can be applied to determine or judge abnormality with respect to an input signal, which can be a useful assistance for medical diagnosis of relevant symptoms.

[0025]

FIG. 1C shows a block diagram of process steps to be performed based on some embodiments of the invention. An input signal is fed into the active learning system 10, an encoder neural network of the active learning system 10 determines features of the input signal in step SS1 and stores the features into a working memory (not shown). Further, an uncertainty measure is determined by a trained neural network 301 of the active learning system 10 in step SS2 and a result of the uncertainty measure is stored in the working memory. The features determined in SS1 is reconstructed by a decoder NN in step SS3 and a reconstructed signal is stored in the working memory. In step SS4, the reconstructed signal is fed from the working memory and compared with the input signal to compute a reconstruction error. The reconstruction error is stored in the working memory and fed to step SS5. In step SS5, the uncertain measure is read from the working memory and combined with the reconstruction error. In step SS6, the input signal is labeled according to a ranking score and the labeled signal is used in step SS7 for training the neural networks in the active learning system 10.

[0026]

FIG. 1D shows a block diagram indicating an active learning process 11 and a convolutional neural network (CNN) training process 21, both of which are performed in the active learning system 10. Upon an identical input signal 12 (or input images 12), the active learning process 11 feeds the input signal 12 to a convolutional neural network (CNN) 13 and the CNN 13 extracts features 14 from the input signal 12. Further, the active learning process 11 computes an uncertainty measure 16 from the features 14 and provides a score 17 based on the uncertainty measure 16.

[0027]

In the CNN training process 21, the input signal 12 is fed to the CNN 13 and the CNN 13 extracts the features 14 from the input signal 12. Then a CNN decoder 25 reconstructs a signal 26 from the features 14 to compare with the input signal 12. By comparing the input signal 12 and the reconstructed signal 26, the CNN training process 21 computes or generates a reconstruction error 27. The active learning system 10 combines the reconstruction error 27 and the uncertainty measure 16, and ranks the input signal 12 by a score 17.

[0028]

When the score 17 is higher than a predetermined threshold, the input signal 12 is fed to a labeling interface (not shown) that allows an operator to annotate the input signal 12 according to one of predetermined classified labels, which is indicated as Human labeling process 18. The process steps performed in the active learning process 11 and the CNN training process 21 described above are illustrated in FIG. 1E, which shows key process steps performed in the active learning system 10.

[0029]

In some embodiments of the invention, the rank is defined based on an addition of an entropy function and the reconstruction error.

[0030]

FIG. 2 shows a block diagram of process steps for ranking the importance of unlabeled images in an active learning system according to some embodiments of the invention. When an input image 103 is provided to a front end of the NN 301 in step 302, the trained NN 301 generates features 303 and outputs a classification result via a softmax output layer 304. The classification result is used for calculating the importance score 104 of the input signal through uncertainty measure 305 based on the Rényi entropy.

[0031]

The trained NN 301 is used for extracting the features 303 for each of the images in the unlabeled dataset 103 and also for computing classifications by the softmax output layer 304. The classification result obtained by the softmax output layer 304 is a probability vector of dimension D where the dimension D is the number of object classes. Denoting the input image by x and the classification result computed by the softmax output layer 304 indicating a probability vector by p , each dimension of the probability vector p represents the probability that the input image 103 belongs to a specific class. The sum of the components of p is equal to one. The uncertainty of the class of the input image can then be measured in the step of the uncertain measure 305 by an entropy function $H(x)$. When the entropy $H(x)$ is computed based on the Shannon entropy, the uncertainty of the class of the input image is given by

$$H(x) = \sum_{i=1}^D -p_i \log p_i \quad (1)$$

[0032]

In an uncertainty method, the uncertainty measure can be used as the importance score of the unlabeled image 104. Further, other entropy measures

defined in the Renyi entropy category can be used for the uncertainty computation.

For instance, the entropy function $H(x)$ may be Collision entropy,

$$H(x) = -\log \sum_{i=1}^D p_i^2 \text{ or Min-entropy, } H(x) = -\log \max_i p_i.$$

[0033]

Further, entropy based methods may be defined by $H(x) = 1 - \log \max_i p_i$ for obtaining an estimate of uncertainty, and an experimental result is shown in FIG. 7.

[0034]

Since the uncertainty method is a universal active learning method, it can be used in conjunction with various classifiers (SVMs, Gaussian processes, or neural networks) as long as the vector representing the class probability can be derived from each input image. In this case, the uncertainty method does not utilize the property of the classifier and reaches sub-optimal performance.

[0035]

In accordance with some embodiments, an approach to improve the uncertainty method by utilizing the property of neural network computation is described in the following. It is established that a neural network computes a hierarchy of feature representation as processing an input image. The completeness of the feature representation can be used to judge how well the neural network models the input image. In order to quantify the completeness of the feature representation, an autoencoder neural network can be used.

[0036]

FIG. 3 shows a block diagram of an autoencoder neural network 710 according to some embodiments of the invention. The autoencoder neural network 710 includes an encoder neural network 701, a decoder neural network 705, and a softmax output layer 703.

[0037]

When an input image 700 is provided, the autoencoder NN 710 outputs classification results 703 from the features 702 extracted by the encoder neural network 701. Further, the features 702 are transmitted to the decoder neural network 705. The decoder neural network 705 generates a reconstructed image 704 from the features 702 extracted by the encoder NN 701. In some cases, the encoder NN 701 may be referred to as a first sub-network #1, and the decoder neural network 705 may be referred to as a second sub-network #2. The first sub-network 701 extracts the features 702 from the input image 700. The extracted features 702 are fed into the softmax output layer 703 that outputs classification results. In this case, the extracted features 702 are also fed into the second sub-network #2. The second sub-network #2 generates a reconstructed image 704 from the features 702 and outputs the reconstruction image.

[0038]

In some embodiments, a reconstruction error is defined based on the Euclidean distance between an input image (or input signal) and a reconstructed image (or reconstructed signal).

[0039]

Further, the reconstructed image 704 is compared to the input image 700 based on the Euclidean distance measurement. The Euclidean distance between the input image 700 and the reconstructed image 704 can be used for quantifying the completeness of the feature representation. When letting x be the vector representation of the input image and y be the vector representation of the reconstructed image, the reconstruction error measure $R(x)$ is defined by the Euclidean distance as follows.

$$R(x) = \|x - y\|_2^2 \quad (2)$$

[0040]

The Euclidean distance indicates how the input image is well represented by the feature representation. When a reconstruction error $R(x)$ is small, it indicates that the neural network models the input image well. However, when the reconstruction error $R(x)$ is large, then it indicates that the neural network does not model the input image well. In some embodiments, including the input image in training improves the representation power (accuracy) of the autoencoder NN 710.

[0041]

For ranking the importance of an input image, the following formula can be used,

$$\alpha H(x) + \beta R(x) \quad (3)$$

where α and β are non-negative weighting parameters.

[0042]

FIG. 4 shows a block diagram indicating an integrated design of sub-networks #1 and #2 used in the uncertainty measure based an active learning system 720 according to some embodiments of the invention. The block diagram shows data process steps used in a method for ranking the importance of unlabeled images in the active learning system 720. The active learning system 720 includes an encoder neural network 701 (first sub-network #1), a softmax output layer 703, a ranking layer 205, a decoder neural network (second sub-network #2).

[0043]

When the input image 700 is provided to the active learning system 720, the encoder NN 701 generates the features 702 from the input image 700. The features 702 can be used for generating a classification result via the Softmax output layer 703. The classification result is fed to the ranking layer 205. Further, the features 702 is fed to the decoder NN 705 and used to generate a reconstructed image 704 by using the decoder NN 705. The reconstructed image 704 is fed to the ranking layer 205. At the ranking layer 205, the classification result and the reconstructed

image are used to compute the importance score 104 with respect to an unlabeled image of the input image 700.

[0044]

The importance score 104 of the unlabeled image can be calculated from the classification output 703 and the reconstructed image 704 by using the ranking layer 205 in the calculation step. After obtaining the importance score 104 regarding the unlabeled image, the active learning system outputs the importance score 104 as an output.

[0045]

FIG. 5 shows a block diagram of an active learning system 600 according to some embodiments of the invention. The active learning system 600 includes a human machine interface (HMI) 610 connectable with a keyboard 611 and a pointing device/medium 612, a processor 620, a storage device 630, a memory 640, a network interface controller 650 (NIC) connectable with a network 690 including local area networks and internet network, a display interface 660, an imaging interface 670 connectable with an imaging device 675, a printer interface 680 connectable with a printing device 685. The processor 620 may include one or more than one central processing unit (CPU). The active learning system 600 can receive electric text/imaging documents 695 via the network 690 connected to the NIC 650. The active learning system 600 can receive annotation data from the annotation device 613 via the HMI 610. Further, the annotation device 613 includes a display screen, and the display screen of the annotation device 613 is configured to display the labeling interface 106 that allows the operator to perform labeling process of unlabeled images stored in the memory 640 by showing the unlabeled image in the display region 601 with the selection area 602 having predetermined annotation boxes and predetermined labeling candidates to be selected.

[0046]

The storage device 630 includes original images 631, a filter system module 632, and a neural network 400. For instance, the processor 620 loads the code of the neural network 400 in the storage 630 to the memory 640 and executes the instructions of the code for implementing the active learning. Further, the pointing device/medium 612 may include modules that read programs stored on a computer readable recording medium.

[0047]

FIG. 6 shows an example of the labeling interface 106 according to some embodiments of the invention. The labeling interface 106 includes a display region 601 and a selection area 602. The labeling interface 106 can be installed in the annotation device 613, which indicates the labeling interface 106 on a display of the annotation device 613. In some cases, the labeling interface 106 can be installed an input/output interface (not shown in the figure) connectable to the human machine interface (HMI) 610 via the network 690. When the labeling interface 106 receives an unlabeled image of the K most important unlabeled images 105 in step S6 of FIG. 1A, the labeling interface 106 shows the unlabeled image on the display region 601. The selection area 602 indicates predetermined candidates for labeling the unlabeled image shown on the display region 601. The labeling interface 106 allows an operator to assign one of selectable annotations indicated in the selection area 602 with respect to the unlabeled image shown in the display region 601. In FIG. 6, the selection area 602 provides selection boxes with predetermined labeling candidates: Dog, Cat, Car, and Plane. As an example, FIG. 6 shows an unlabeled image indicating a cat image 603 displayed in the display region 601. In this case, the annotation box of Cat is checked by the operator (annotator) in response to the cat image shown in the selection area 602. The labeling interface 106 is configured to load and show unlabeled images stored

the labeling storage in the memory according to the operations by the operator. The images labeled by the labeling interface 106 are stored into a new training image storage area in the memory in step S3 as newly labeled training images 107 as seen in FIG. 1A.

[0048]

FIG. 7 shows experimental results of image classifications using the active learning methods on a convolutional neural network (CNN) for comparison, and the uncertainty method based on a CANN.

[0049]

For comparison, the following convolutional neural network (CNN) was used for the experiments in the MNIST dataset: (20)5c-2p-(50)5c-2p-500fc-r-10fc, where “(20)5c” denotes a convolutional layer of 20 neurons with a kernel size 5, “2p” denotes a 2×2 pooling, “r” denotes rectified-linear units (ReLU), and “500fc” denotes a fully connected layer with 500 nodes. One softmax loss layer is added to the classification output “10fc” for the backpropagation. For the convolutional autoencoder neural network (CANN) part, the structure from the deconvolutional network is adapted. For the CIFAR10 dataset: “(32)3c-2p-r-(32)3c-r-2p-(64)3c-r-2p-200fc-10fc”. For the CANN part, the structure is the same as mentioned in MNIST settings.

[0050]

In FIG. 7, the dataset “Uncertain. meas. & Recon.” indicates data obtained by the uncertainty measure and reconstruction method according to an embodiment of the invention. The methods other than the uncertainty method shown in FIG. 7 are obtained by using a CNN instead of the structure with an autoencoder. Further, “RDM” indicates random method, “EMC” indicates an expected model change method, “UNC” indicates an uncertainty method without reconstruction, “DW” indicates a density weighted method, and “FF” indicates a farthest first method. In

both MNIST setting and CIFAR10 setting, the uncertainty measure & reconstruction method in accordance with the embodiment of the invention shows superior performance compared to the other methods. This indicates one of advantages of the active learning system in accordance with some embodiments of the invention.

[0051]

The advantage is reducing the number of annotated data, as discussed above, the artificial neural network according to some embodiments of the invention can provide less annotation processes with improving the classification accuracy, the use of artificial neural network that determines an uncertainty measure may reduce central processing unit (CPU) usage, power consumption, and/or network bandwidth usage, which is advantageous for improving the functioning of a computer.

[0052]

The above-described embodiments of the present invention can be implemented in any of numerous ways. For example, the embodiments may be implemented using hardware, software or a combination thereof. When implemented in software, the software code can be executed on any suitable processor or collection of processors, whether provided in a single computer or distributed among multiple computers. Such processors may be implemented as integrated circuits, with one or more processors in an integrated circuit component. Though, a processor may be implemented using circuitry in any suitable format. The processor can be connected to memory, transceiver, and input/output interfaces as known in the art.

[0053]

Also, the various methods or processes outlined herein may be coded as software that is executable on one or more processors that employ any one of a

variety of operating systems or platforms. Alternatively, or additionally, the invention may be embodied as a computer readable medium other than a computer-readable storage medium, such as signals.

[0054]

The terms "program" or "software" are used herein in a generic sense to refer to any type of computer code or set of computer-executable instructions that can be employed to program a computer or other processor to implement various aspects of the present invention as discussed above.

[0055]

Use of ordinal terms such as "first," "second," in the claims to modify a claim element does not by itself connote any priority, precedence, or order of one claim element over another or the temporal order in which acts of a method are performed, but are used merely as labels to distinguish one claim element having a certain name from another element having a same name (but for use of the ordinal term) to distinguish the claim elements.

[0056]

Although several preferred embodiments have been shown and described, it would be apparent to those skilled in the art that many changes and modifications may be made thereunto without the departing from the scope of the invention, which is defined by the following claims and their equivalents.