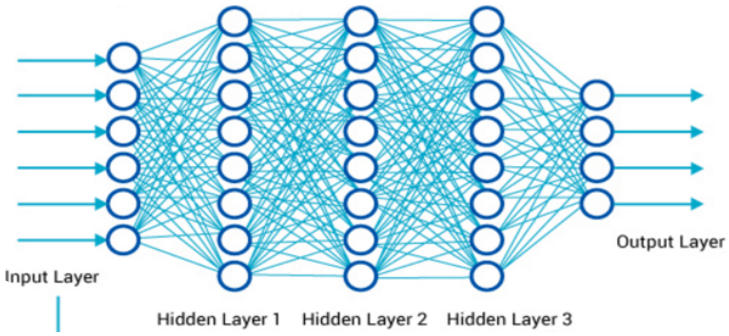
ICT-6522

Data Warehousing and Mining

Term Paper

Deep Learning for Pattern Recognition



Submitted to

Dr. Hossen Asiful Mustafa

PhD, University of South Carolina, USA

Assistant Professor, Institute of Information and Communication Technology (IICT)

Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh

Submitted By

Md Abdul Aowal

Student No.: **1015312015**

Submitted On

05/10/2016

Deep Learning for Pattern Recognition

Md Abdul Aowal
Institute of Information and Communication Technology (IICT)
Bangladesh University of Engineering and Technology, Dhaka-1205, Bangladesh
E-mail: aowal.eee@gmail.com

***Abstract*-Deep learning has emerged as a new area of machine learning research. During the past several years, the techniques developed from deep learning research have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech work within the traditional and the new, widened scopes including key aspects of machine learning and artificial intelligence. Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. In addition, Deep learning has become more useful as the amount of available training data has increased. In this paper, we have focused on few important applications of deep learning in various fields that has been implemented in the last few years and also discussed some possible future applications.**

***Keywords*—Deep learning, Convolutional Neural Network, Recurrent Neural Networks, Unsupervised learning**

I. INTRODUCTION

Inventors have long been dreaming of creating machines that think. This desire dates back to at least the time of ancient Greece. The mythical figures Pygmalion, Daedalus, and Hephaestus may all be interpreted as legendary inventors, and Galatea, Talos, and Pandora may all be regarded as artificial life. When programmable computers were first conceived, people wondered whether they might become intelligent, over a hundred years before one was built (Lovelace, 1842). Today, artificial intelligence (AI) is a thriving field with many practical applications and active research topics. We look to intelligent software to automate routine labor, understand speech or images, make diagnoses in medicine and support basic scientific research.

In the early days of artificial intelligence, the field rapidly tackled and solved problems that are intellectually difficult for human beings but relatively straight-forward for computers—problems that can be described by a list of formal, mathematical rules. The true challenge to artificial intelligence proved to be solving the tasks that are easy for people to perform but hard for people to describe formally—problems that we solve intuitively, that feel automatic, like recognizing spoken words or faces in images.

Deep learning is about a solution to these more intuitive problems. This solution is to allow computers to learn from experience and understand the world in terms of a hierarchy of concepts, with each concept defined in terms of its relation to simpler concepts. By gathering knowledge from experience, this approach avoids the need for human operators to formally specify all of the knowledge that the computer needs. The hierarchy of concepts allows the computer to learn complicated concepts by building them out of simpler ones. If we draw a graph showing how these concepts are built on top of each other, the graph is deep, with many layers. For this reason, this approach often calls as AI deep learning.

Many of the early successes of AI took place in relatively sterile and formal environments and did not require computers to have much knowledge about the world. For example, IBM's Deep Blue chess-playing system defeated world champion Garry Kasparov in 1997. Chess is of course a very simple world, containing only sixty-four locations and thirty-two pieces that can move in only rigidly circumscribed ways. Devising a successful chess strategy is a tremendous accomplishment, but the challenge is not due to the difficulty of describing the set of

chess pieces and allowable moves to the computer. Chess can be completely described by a very brief list of completely formal rules, easily provided ahead of time by the programmer.

Ironically, abstract and formal tasks that are among the most difficult mental undertakings for a human being are among the easiest for a computer. Computers have long been able to defeat even the best human chess player, but are only recently matching some of the abilities of average human beings to recognize objects or speech. A person's everyday life requires an immense amount of knowledge about the world. Much of this knowledge is subjective and intuitive, and therefore difficult to articulate in a formal way. Computers need to capture this same knowledge in order to behave in an intelligent way. One of the key challenges in artificial intelligence is how to get this informal knowledge into a computer.

Machine-learning technology powers many aspects of modern society: from web searches to content filtering on social networks to recommendations on e-commerce websites, and it is increasingly present in consumer products such as cameras and smartphones. Machine-learning systems are used to identify objects in images, transcribe speech into text, match news items, posts or products with users' interests, and select relevant results of search. Increasingly, these applications make use of a class of techniques called deep learning.

Deep learning is making major advances in solving problems that have resisted the best attempts of the artificial intelligence community for many years. It has turned out to be very good at discovering intricate structures in high-dimensional data and is therefore applicable to many domains of science, business and government. In addition, to beating records in image recognition [1,2] has beaten other machine-learning techniques at predicting the activity of potential drug molecules [4], reconstructing brain circuits [5], and predicting the effects of mutations in non-coding DNA on gene expression and disease [6,7]. Perhaps more surprisingly, deep learning has produced extremely promising results for various tasks in natural language understanding [8], particularly topic classification, sentiment analysis, question answering [9].

We think that deep learning will have many more successes in the near future in pattern classifaction because it requires very little engineering by hand, so it can easily take advantage of increases in the amount of available computation and data. New learning algorithms and architectures that are currently being developed for deep neural networks will only accelerate this progress.

II. SUPERVISED LEARNING: A BRIEF REVIEW

The most common form of machine learning, deep or not, is supervised learning. Imagine that we want to build a system that can classify images as containing, say, a house, a car, a person or a pet. We first collect a large data set of images of houses, cars, people and pets, each labelled with its category. During training, the machine is shown an image and produces an output in the form of a vector of scores, one for each category. We want the desired category to have the highest score of all categories, but this is unlikely to happen before training. We compute an objective function that measures the error (or distance) between the output scores and the desired pattern of scores. The machine then modifies its internal adjustable parameters to reduce this error. These adjustable parameters, often called weights, are real numbers that can be seen as 'knobs' that define the input–output function of the machine. In a typical deep-learning system, there may be hundreds of millions of these adjustable weights, and hundreds of millions of labelled examples with which to train the machine.

To properly adjust the weight vector, the learning algorithm computes a gradient vector that, for each weight, indicates by what amount the error would increase or decrease if the weight were increased by a tiny amount. The weight vector is then adjusted in the opposite direction to the gradient vector.

The objective function, averaged over all the training examples, can be seen as a kind of hilly landscape in the high-dimensional space of weight values. The negative gradient vector indicates the direction of steepest descent in this landscape, taking it closer to a minimum, where the output error is low on average.

In practice, most practitioners use a procedure called stochastic gradient descent (SGD). This consists of showing the input vector for a few examples, computing the outputs and the errors, computing the average gradient for those examples, and adjusting the weights accordingly. The process is repeated for many small sets of examples from the training set until the average of the objective function stops decreasing. It is called stochastic because each small set of examples gives a noisy estimate of the average gradient over all examples. This simple procedure usually finds a good set of weights surprisingly quickly when compared with far more elaborate optimization techniques [10]. After training, the performance of the system is measured on a different set of examples called a test set. This serves to test the generalization ability of the machine — its ability to produce sensible answers on new inputs that it has never seen during training.

Many of the current practical applications of machine learning use linear classifiers on top of hand-engineered features. A two-class linear classifier computes a weighted sum of the feature vector components. If the weighted sum is above a threshold, the input is classified as belonging to a particular category. Here, linear classifiers can only carve their input space into very simple regions, namely half-spaces separated by a hyperplane [11]. But, problems such as image and speech recognition require the input–output function to be insensitive to irrelevant variations of the input, such as variations in position, orientation or illumination of an object, or variations in the pitch or accent of speech, while being very sensitive to particular minute. A linear classifier, or any other 'shallow' classifier operating on raw pixels could not possibly distinguish the latter two, while putting the former two in the same category. This is why shallow classifiers require a good feature extractor that solves the selectivity–invariance dilemma — one that produces representations that are selective to the aspects of the image that are important for discrimination, but that are invariant to irrelevant aspects such as the pose of the animal.

To make classifiers more powerful, one can use generic non-linear features, as with kernel methods, but generic features such as those arising with the Gaussian kernel do not allow the learner to generalize well far from the training examples. The conventional option is to hand design good feature extractors, which requires a considerable amount of engineering skill and domain expertise. But this can all be avoided if good features can be learned automatically using a general-purpose learning procedure. This is the key advantage of deep learning.

A deep-learning architecture is a multilayer stack of simple modules, all (or most) of which are subject to learning, and many of which compute non-linear input–output mappings. Each module in the stack transforms its input to increase both the selectivity and the invariance of the representation.

III. Backpropagation Technique: A solution to greedy approach

To replace hand-engineered features with trainable multilayer networks, but despite its simplicity multilayer architectures can be trained by simple stochastic gradient descent. As long as the modules are relatively smooth functions of their inputs and of their internal weights, one can compute gradients using the backpropagation procedure. The backpropagation procedure to compute the gradient of an objective function with respect to the weights of a multilayer stack of modules is nothing more than a practical application of the chain rule for derivatives. The key insight is that the derivative (or gradient) of the objective with respect to the input of a module can be computed by working backwards from the gradient with respect to the output of that module.

Many applications of deep learning use feedforward neural network architectures (Fig. 1), which learn to map a fixed-size input (for example, an image) to a fixed-size output (for example, a probability for each of several categories). To go from one layer to the next, a set of units compute a weighted sum of their inputs from the previous layer and pass the result through a non-linear function. Units that are not in the input or output layer are conventionally called hidden units. The hidden layers can be seen as distorting the input in a non-linear way so that categories become linearly separable by the last layer (Fig. 1).

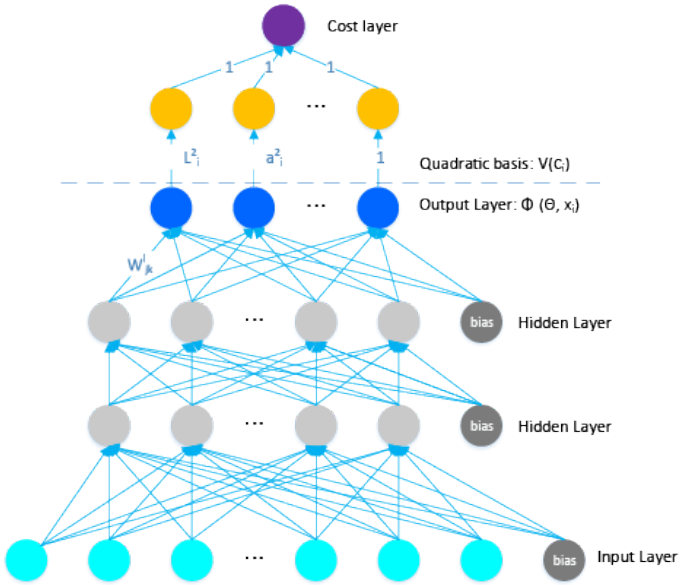


Figure 1. A deep neural network consists of hidden layers. Note that the weights for the connections between the blue neurons and the yellow neurons are just the elements of the quadratic color basis, and the activation function in the yellow and purple neurons is the identity function. During training, error backpropagation starts from the output layer, as the connection weights above the dash line have already been fixed.

In particular, simple gradient descent get trapped in poor local minima — weight configurations for which no small change would reduce the average error. In practice, poor local minima are rarely a problem with large networks. Regardless of the initial conditions, the system nearly always reaches solutions of very similar quality. Recent theoretical and empirical results strongly suggest that local minima are not a serious issue in general.

The objective in learning each layer of feature detectors was to be able to reconstruct or model the activities of feature detectors (or raw inputs) in the layer below. By 'pre-training' several layers of progressively more complex feature detectors using this reconstruction objective, the weights of a deep network could be initialized to sensible values. A final layer of output units could then be added to the top of the network and the whole deep system could be fine-tuned using standard backpropagation [12]. This worked remarkably well for recognizing handwritten digits or for detecting pedestrians, especially when the amount of labelled data was very limited [13].

There was, however, one particular type of deep, feedforward network that was much easier to train and generalized much better than networks with full connectivity between adjacent layers. This was the convolutional neural network (ConvNet) [14,15].

IV. Convolutional Neural Networks

A. ConvNet: A Brief Overview

A convolutional neural network, or preferably convolutional network or ConvNets uses convolutional layers that filter inputs for useful information. These convolutional layers have parameters that are learned so that these filters are adjusted automatically to extract the most useful information for the task at hand.

For example, in a general object recognition task it might be most useful to filter information about the shape of an object (objects usually have very different shapes) while for a bird recognition task it might be more suitable to extract information about the color of the bird (most birds have a similar shape, but different colors; here color is more useful to distinguish between birds). Convolutional networks adjust automatically to find the best feature for these tasks. Many data modalities are in the form of multiple arrays: 1D for signals and sequences, including language; 2D for images or audio spectrograms; and 3D for video or volumetric images. There are four key ideas behind ConvNets that take advantage of the properties of natural signals: local connections, shared weights, pooling and the use of many layers.

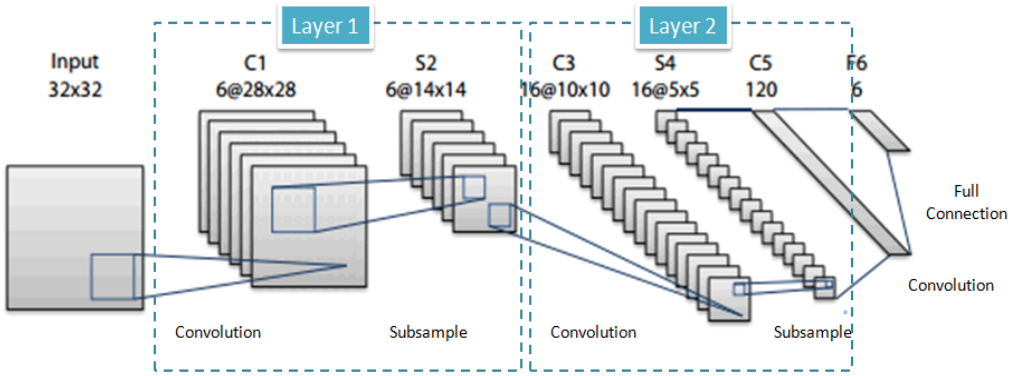


Figure 2a. A complete process of convolutional neural network consist of convolution and subsampling.

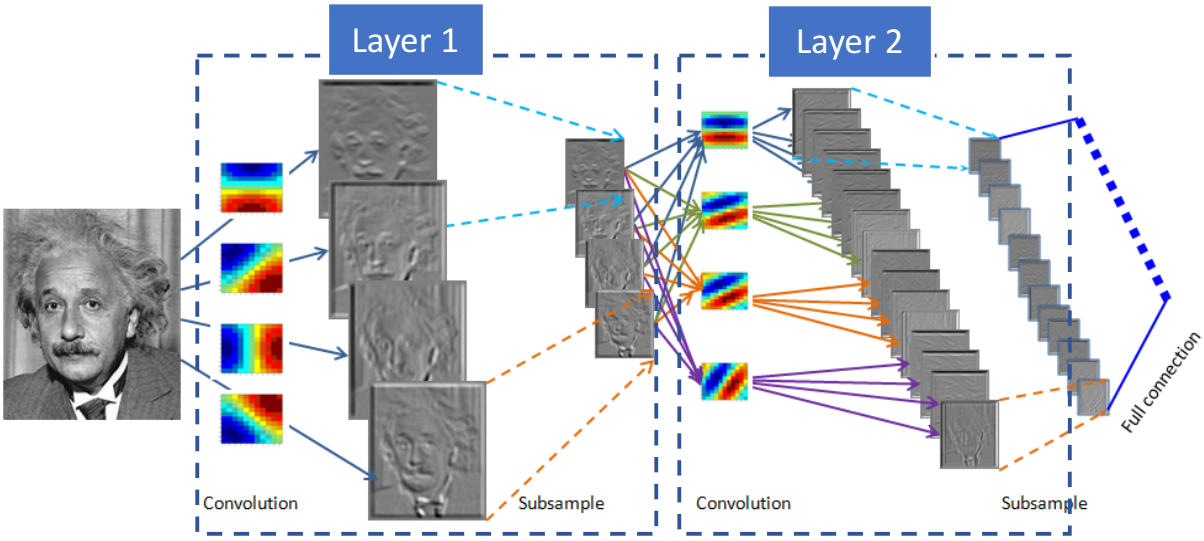


Figure 2b. An image of a Einstein is filtered by 4 convolutional kernels which create 4 feature maps, these feature maps are subsampled by max pooling. The next layer applies 12 convolutional kernels to these subsampled images and again we pool the feature maps. The final layer is a fully connected layer where all generated features are combined and used in the classifier.

The architecture of a typical ConvNet (Fig. 2) is structured as a series of stages. The first few stages are composed of two types of layers: convolutional layers and pooling layers. Units in a convolutional layer are organized in feature maps, within which each unit is connected to local patches in the feature maps of the previous layer through a set of weights called a filter bank. The result of this local weighted sum is then passed through a non-linearity. All units in a feature map share the same filter bank. Different feature maps in a layer use different filter banks. The reason for this architecture is twofold. First, in array data such as images, local groups of values are often highly correlated, forming distinctive local motifs that are easily detected. Second, the local statistics of images and other signals are invariant to location. In other words, if a motif can appear in one part of the image,

it could appear anywhere, hence the idea of units at different locations sharing the same weights and detecting the same pattern in different parts of the array. Mathematically, the filtering operation performed by a feature map is a discrete convolution, hence the name.

Although the role of the convolutional layer is to detect local con- junctions of features from the previous layer, the role of the pooling layer is to merge semantically similar features into one. Because the relative positions of the features forming a motif can vary somewhat, reliably detecting the motif can be done by coarse-graining the position of each feature. A typical pooling unit computes the maximum of a local patch of units in one feature map (or in a few feature maps). Neighboring pooling units take input from patches that are shifted by more than one row or column, thereby reducing the dimension of the representation and creating an invariance to small shifts and distortions. Two or three stages of convolution, non-linearity and pooling are stacked, followed by more convolutional and fully-connected layers. Backpropagating gradients through a ConvNet is as simple as through a regular deep network, allowing all the weights in all the filter banks to be trained.

Deep neural networks exploit the property that many natural signals are compositional hierarchies, in which higher-level features are obtained by composing lower-level ones. In images, local combinations of edges form motifs, motifs assemble into parts, and parts form objects. Similar hierarchies exist in speech and text from sounds to phones, phonemes, syllables, words and sentences. The pooling allows representations to vary very little when elements in the previous layer vary in position and appearance.

B. Applications in image understanding

ConvNets have been applied with great success to the detection, segmentation and recognition of objects and regions in images. These were all tasks in which labelled data was relatively abundant, such as traffic sign recognition [16], the segmentation of biological images [17] particularly for connect omics, and the detection of faces, text, pedestrians and human bodies in natural images. A major recent practical success of ConvNets is face recognition [18].

Importantly, images can be labelled at the pixel level, which will have applications in technology, including autonomous mobile robots and self-driving cars. Companies such as Mobileye and NVIDIA are using such ConvNet-based methods in their upcoming vision systems for cars. Other applications gaining importance involve natural language understanding [8] and speech recognition [19].

Recent ConvNet architectures have 10 to 20 layers, hundreds of millions of weights, and billions of connections between units. Whereas training such large networks could have taken weeks only two years ago, progress in hardware, software and algorithm parallelization have reduced training times to a few hours. The performance of ConvNet-based vision systems has caused most major technology companies, including Google, Facebook, Microsoft, IBM, Yahoo!, Twitter and Adobe, as well as a quickly growing number of start-ups to initiate research and development projects and to deploy ConvNet-based image understanding products and services.

ConvNets are easily amenable to efficient hardware implementations in chips or field-programmable gate arrays. A number of companies such as NVIDIA, Mobileye, Intel, Qualcomm and Samsung are developing ConvNet chips to enable real-time vision applications in smartphones, cameras, robots and self-driving cars.

C. Applications in image understanding : Image Caption Generator

Despite these successes, ConvNets were largely forsaken by the mainstream computer-vision and machine-learning communities until the ImageNet competition in 2012. When deep convolutional networks were applied to a data set of about a million images from the web that contained 1,000 different classes, they achieved

spectacular results, almost halving the error rates of the best competing approaches [1]. This success came from the efficient use of GPUs, ReLUs, a new regularization technique called dropout, and techniques to generate more training examples by deforming the existing ones. This success has brought about a revolution in computer vision; ConvNets are now the dominant approach for almost all recognition and detection tasks and approach human performance on some tasks. A recent stunning demonstration combines ConvNets and recurrent net modules for the generation of image captions (Fig. 3).

Figure 3. **From image to text.** It generates complete sentences in natural language extracted by a deep convolution neural network (CNN) an input image from a test image, with the RNN trained to 'translate' high-level representations of images into captions (top). A selection of evaluation results grouped by human rating (bottm).

They followed an elegant recipe, replacing the commonly used encoder RNN by a deep convolution neural network (CNN). Over the last few years it has been convincingly shown that CNNs can produce a rich representation of the input image by embedding it to a fixed-length vector, such that this representation can be used for a variety of vision tasks. Hence, it is natural to use a CNN as an image "encoder", by first pre-training it for an image classification task and using the last hidden layer as an input to the RNN decoder that generates sentences (see Fig. 3).

Their main contributions are as follows. First, they presented an end-to-end system for the problem (NIC). It is a neural net which is fully trainable using stochastic gradient descent. Second, they explained a model combines state-of-art sub-networks for vision and language models. These can be pre-trained on larger corpora and thus can take advantage of additional data. Finally, it yields significantly better performance compared to state-of-the-art approaches; for instance, on the Pascal dataset, NIC yielded a BLEU score of 59, to be compared to the current state-of-the-art of 25, while human performance reaches 69. On Flickr30k, they improved from 56 to 66, and on SBU, from 19 to 28.

NIC, an end-to-end neural network system that can automatically view an image and generate a reasonable description in plain English. NIC is based on a convolution neural network that encodes an image into a compact representation, followed by a recurrent neural net- work that generates a corresponding sentence. The model is trained to maximize the likelihood of the sentence given the image. Experiments on several datasets show the robustness of NIC in terms of qualitative results (see Fig. 3) and quantitative evaluations, using either ranking metrics or BLEU, a metric used in ma- chine translation to evaluate the quality of generated sentences. It is clear from these experiments that, as the size of the available datasets for image description increases, so will the performance of approaches like NIC.

D. Applications in image classifications : PCAnet

Another interesting CNN based image classification framework has been developed by Tsung-Han Chan et al. [20]. The proposed method is a simple but effective baseline for deep learning. They propose a novel two-layer architecture where each layer convolves the image with a filter bank, followed by binary hashing, and finally block histogramming for indexing and pooling. The filters in the filter bank are learned using simple algorithms such as random projections (RandNet), principal component analysis (PCANet), and linear discriminant analysis (LDANet). They report results competitive with those obtained by other deep learning methods and scattering networks on a variety of task: face recognition, face verification, hand-written digit recognition, texture discrimination, and object recognition.

The main algorithm cascades two filter bank convolutions with an intermediate mean normalization step, followed by a binary hashing step and a final histogramming step. Training involves estimating the filter banks used for the convolutions, and estimating the classifier to be used on top of the ultimate histogram-derived features (see Fig 4).

The authors estimate a multiclass linear SVM to operate on the estimated feature vector for each image. The same setup was used for all input data. The particular SVM implementation was Liblinear. The specific algorithm used was l_2-regularized l_2-loss support vector one-against-rest support vector classification and a cost of 1.

According to this paper, they have proposed arguably the simplest unsupervised convolutional deep learning network— PCANet. The network processes input images by cascaded PCA, binary hashing, and block histograms. Like the most ConvNet models, the network parameters such as the number of layers, the filter size, and the number of filters have to be given to PCANet. Once the parameters are fixed, training PCANet is extremely simple and efficient, for the filter learning in PCANet does not involve regularized parameters and does not require numerical optimization solver. Moreover, building the PCANet comprises only a cascaded linear map, followed by a nonlinear output stage. Such a simplicity offers an alternative and yet refreshing perspective to convolutional deep learning networks, and could further facilitate mathematical analysis and justification of its effectiveness.

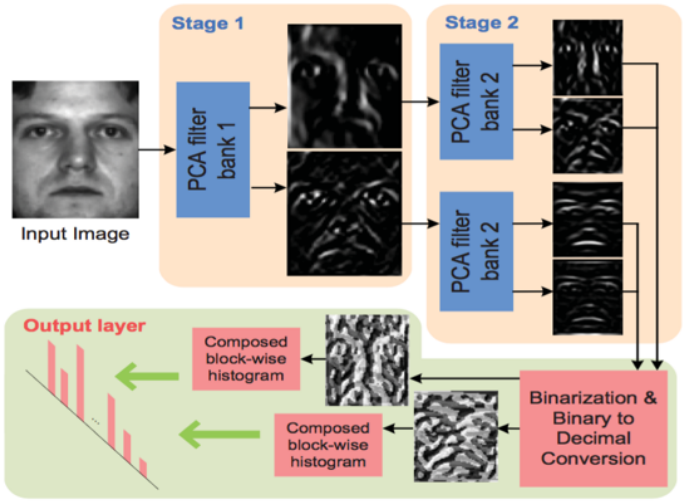


Figure 4. Illustration of how the proposed PCANet extracts features from an image through three simplest processing components: PCA filters, binary hashing, and histogram.

A couple of simple extensions of PCANet; that is, RandNet and LDANet, have been introduced and tested together with PCANet on many image classification tasks, including face, hand-written digit, texture, and object. Extensive experimental results have consistently shown that the PCANet outperforms RandNet and LDANet, and is generally on par with ScatNet and variations of ConvNet. Furthermore, the performance of PCANet is closely comparable and often better than highly engineered hand-crafted features (such as LBP and LQP). In tasks such as face recognition, PCANet also demonstrates remarkable robustness to corruption and ability to transfer to new datasets.

The experiments also convey that as long as the images in databases are somehow well prepared; i.e., images are roughly aligned and do not exhibit diverse scales or poses, PCANet is able to eliminate the image variability and gives reasonably competitive accuracy. In challenging image databases such as Pascal and ImageNet, PCANet might not be sufficient to handle the variability, given its extremely simple structure and unsupervised learning method. An intriguing research direction will then be how to construct a more complicated (say more sophisticated filters possibly with discriminative learning) or deeper (more number of stages) PCANet that could accommodate the aforementioned issues. The current bottleneck that keeps PCANet from growing deeper (e.g., more than two stages) is that the dimension of the resulted feature would increase ex potentially with the number of stages.

Regardless, extensive experiments given in this paper sufficiently conclude two facts. First, the PCANet is a very simple deep learning network, effectively extracting useful information for classification of faces, digits, and texture images; second, the PCANet can be a valuable baseline for studying advanced deep learning architectures for large-scale image classification tasks.

D. Applications in pictorial aesthetics rating : RAPID

Aesthetic quality classification plays an important role in our day to day life. Aesthetic generally refers to the beauty or the appreciation of beauty. The aesthetic beauty of a picture is determined by many factors, some of these aesthetic quality factors include cuteness, prettiness, messiness, neatness, cuddleness, loveliness, organized, disorganized. It creates radical change in the sweet aesthetic sensations in both physiological and psychological. Through a quality image we can see into the life of things. Specially, in fine art, especially painting, humans have mastered the skill to create unique visual experiences through composing a complex interplay between the con-

tent and style of an image. Thus far the algorithmic basis of this process is unknown and there exists no artificial system with similar capabilities.

Xin Lu et al. [21] presented a new method named RAPID (RAting PIctorial aesthetics using Deep learn- ing) system, which adopts a novel deep neural network approach to enable automatic feature learning. The central idea is to incorporate heterogeneous inputs generated from the image, which include a global view and a local view, and to unify the feature learning and classifier training using a double-column deep convolutional neural network. In addition, they utilized the style attributes of images to help improve the aesthetic quality categorization accuracy.

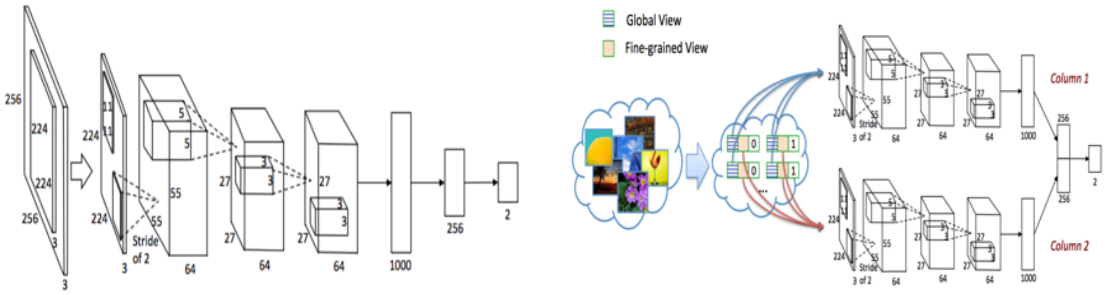


Figure 5. In the left, single-column convolutional neural network for aesthetic quality rating and categorization. Four convolutional layers and two fully- connected layers. The first and second convolutional layers are followed by max-pooling layers and normalization layers. In the right, double-column convolutional neural network. Each training image is represented by its global and local views, and is associated with its aesthetic quality label: 0 refers to a low quality image and 1 refers to a high quality image.

According to the paper, they explored the two natural ways to formulate the problem. The first is to leverage the idea of multi-task learning, which jointly construct feature representation and minimize the classification error for both labels. They postulated the problems as an optimization problem assuming aesthetic quality labels $\{y_{ai}\}$ and style labels $\{y_{si}\}$ for all training images:

$$\max_{\mathbf{X},\mathbf{W}_a,\mathbf{W}_s} \sum_{i=1}^{N}\Big(\sum_{c\in C_A}\mathbb{I}(y_{ai}=c)\log p(y_{ai}\mid \mathbf{x}_i,\mathbf{w}_{ac})+\sum_{c\in C_S}\mathbb{I}(y_{si}=c)\log p(y_{si}\mid \mathbf{x}_i,\mathbf{w}_{sc})\Big)$$

where **X** is the features of all training images, $\boldsymbol{C_A}$ is the label set for aesthetic quality, $\boldsymbol{C_S}$ is the label set for style, and $\mathbf{W_a} = \{\mathbf{w_{ac}}\}c\in C_A$ and $\mathbf{Ws} = \{\mathbf{w_{sc}}\}c\in C_S$ are the model parameters. To facilitate the network training with style attributes of images, they proposed a regularized double-column convolutional neural network (RDCNN) with two normalized inputs of the aesthetic column same as in DCNN. The training of RDCNN is done by solving the following optimization problem:

$$\max_{\mathbf{x}_a,\mathbf{W}_a} \sum_{i=1}^{N}\sum_{c=1\in C_a}\mathbb{I}(y_{ai}=c)\log p(y_{ai}\mid \mathbf{x}_{ai},\mathbf{x}_{si},\mathbf{w}_{ac})$$

where $\mathbf{x_{si}}$ are the style attributes of the **i**-th training image, $\mathbf{x_{ai}}$ are the features to be learned.

In this paper, they explored beyond generic image features by learning effective aesthetics features from images directly. The deep convolutional neural network takes pixels as inputs and learns a suitable representation through multiple convolutional and fully connected layers. Image aesthetics relies on a combination of local and global visual cues. For example, the rule of thirds is a global image cue while sharpness and noise levels are local visual characteristics. Given an image, two heterogeneous inputs generated to represent its global cues and local cues respectively. Figure 5 illustrates global vs. local views. They also developed a double-column neural network

structure which takes parallel inputs from the two columns. One column takes a global view of the image and the other column takes a local view of the image. They also integrated the two columns after some layers of transformations to form the final classifier. Then, further improved the aesthetic quality categorization by exploring style attributes associated with images (see Fig. 6a and 6b). The system is named as RAPID, which stands for RAting PIctorial aesthetics using Deep learning.

This paper demonstrated the effectiveness of style attributes by comparing the best aesthetic quality categorization accuracy they have achieved with and without style attributes. In summary, RDCNN outperforms DCNN for both δ values they experimented with.

(a) Images ranked the highest in aesthetics by DCNN

(b) Images ranked the lowest in aesthetics by DCNN

Figure 6. Examples of images ranked the highest and the lowest in aesthetics generated by Double Column Neural Network (DCNN). Differences between low-aesthetic images and high-aesthetic images heavily lie in the amount of textures and complexity of the whole image.

V. Recurrent neural networks(RNNs)

A. What Are RNNs?

Humans don't start their thinking from scratch every second. As you read this essay, you understand each word based on your understanding of previous words. You don't throw everything away and start thinking from scratch again. Your thoughts have persistence. Traditional neural networks can't do this, and it seems like a major shortcoming. For example, imagine you want to classify what kind of event is happening at every point in a movie. It's unclear how a traditional neural network could use its reasoning about previous events in the film to inform later ones.

Recurrent neural networks address this issue. The idea behind RNNs is to make use of sequential information. In a traditional neural network, we assume that all inputs (and outputs) are independent of each other. But for many tasks that's a very bad idea. If you want to predict the next word in a sentence you better know which words came before it. RNNs are called *recurrent* because they perform the same task for every element of a sequence, with the output being depended on the previous computations. Another way to think about RNNs is that they have a "memory" which captures information about what has been calculated so far. In theory RNNs can make use of information in arbitrarily long sequences, but in practice they are limited to looking back only a few steps (more on this later). (See Fig. 7)

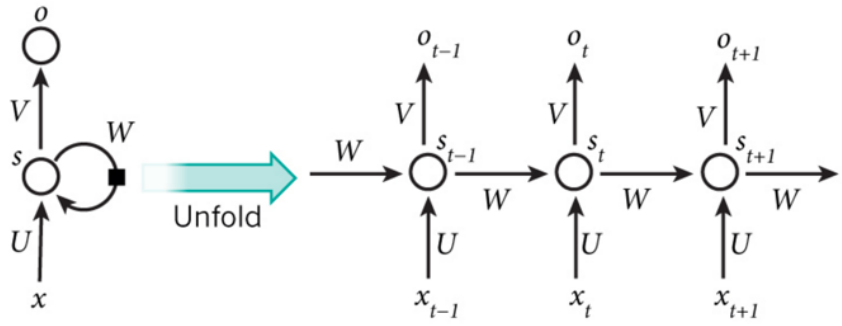


Figure 7. A recurrent neural network and the unfolding in time of the computation involved in its forward computation. The artificial neurons (for example, hidden units grouped under node s with values st at time t) get inputs from other neurons at previous time steps (this is represented with the black square, representing a delay of one time step, on the left). In this way, a recurrent neural network can map an input sequence with elements xt into an output sequence with elements ot, with each ot depending on all the previous xt' (for $t' \leq t$). The same parameters (matrices U,V,W) are used at each time step.

When backpropagation was first introduced, its most exciting use was for training recurrent neural networks (RNNs). For tasks that involve sequential inputs, such as speech and language, it is often better to use RNNs. RNNs process an input sequence one element at a time, maintaining in their hidden units a 'state vector' that implicitly contains information about the history of all the past elements of the sequence.

B. LSTM Networks and theit uses

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn! All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. (see Fig. 8).

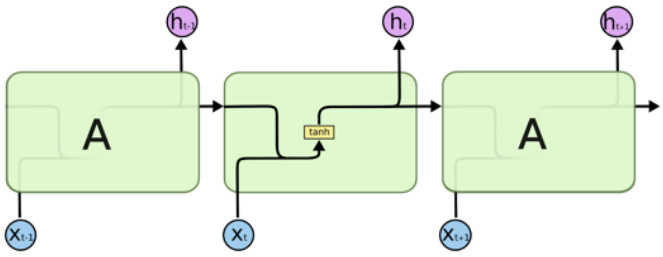


Figure 8. The repeating module in a standard RNN contains a single layer.

LSTM networks have subsequently proved to be more effective than conventional RNNs, especially when they have several layers for each time step, enabling an entire speech recognition system that goes all the way from acoustics to the sequence of characters in the transcription. LSTM networks or related forms of gated units are also currently used for the encoder and decoder networks that perform so well at machine translation.

Over the past year, several authors have made different proposals to augment RNNs with a memory module. Proposals include the Neural Turing Machine in which the network is augmented by a 'tape-like' memory that the RNN can choose to read from or write to, and memory networks, in which a regular network is augmented by a kind of associative memory. Memory networks has yielded excellent performance on standard question-answering benchmarks. The memory is used to remember the story about which the network is later asked to answer questions.

VI. FUTURE WORK: PROBABLE ASPECTS

While deep learning has been successfully applied to challenging pattern inference tasks, the goal of the field is far beyond task-specific applications. This scope may make the comparison of various methodologies increasingly complex and will likely necessitate a collaborative effort by the research community to address. It should also be noted that, despite the great prospect offered by deep learning technologies, some domain-specific tasks may not be directly improved by such schemes. An example is identifying and reading the routing numbers at the bottom of bank checks. Though these digits are human readable, they are comprised of restricted character sets which specialized readers can recognize flawlessly at very high data rates [22]. Similarly, iris recognition is not a task that humans generally perform; indeed, without training, one iris looks very similar to another to the untrained eye, yet engineered systems can produce matches between candidate iris images and an image database with high precision and accuracy to serve as a unique identifier [23]. Furthermore, deep learning platforms can also benefit from engineered features while learning more complex representations which engineered systems typically lack.

Unsupervised learning had a catalytic effect in reviving interest in deep learning, but has since been overshadowed by the successes of purely supervised learning. Although we have not focused on it in this Review, we expect unsupervised learning to become far more important in the longer term. Human and animal learning is largely unsupervised: we discover the structure of the world by observing it, not by being told the name of every object.

Human vision is an active process that sequentially samples the optic array in an intelligent, task-specific way using a small, high-resolution fovea with a large, low-resolution surround. We expect much of the future progress in vision to come from systems that are trained end-to-end and combine ConvNets with RNNs that use reinforcement learning to decide where to look. Systems combining deep learning and reinforcement learning are

in their infancy, but they already outperform passive vision systems at classification tasks and produce impressive results in learning to play many different video games.

Natural language understanding is another area in which deep learning is poised to make a large impact over the next few years. We expect systems that use RNNs to understand sentences or whole documents will become much better when they learn strategies for selectively attending to one part at a time.

Ultimately, major progress in artificial intelligence will come about through systems that combine representation learning with complex reasoning. Although deep learning and simple reasoning have been used for speech and handwriting recognition for a long time, new paradigms are needed to replace rule-based manipulation of symbolic expressions by operations on large vectors.

Despite the myriad of open research issues and the fact that the field is still in its infancy, it is abundantly clear that advancements made with respect to developing deep machine learning systems will undoubtedly shape the future of machine learning and artificial intelligence systems in general.

VII. CONCLUSION

Deep learning has attracted a lot of attention because it is particularly good at a type of learning that has the potential to be very useful for real-world applications. In addition, deep learning networks can be successfully applied to big data for knowledge discovery, knowledge application, and knowledge-based prediction. In other words, deep learning can be a powerful engine for producing actionable results. The discovery and recognition of patterns and regularities in the world around us lies at the heart of scientific and technological progress. It's how we advance and how we innovate. It's also an area where deep learning excels. The question isn't whether or not deep learning is useful, it's how can we use deep learning to improve what we're already doing, or to gain new insights from the data we already have.

ACKNOWLEDGMENT

First of all, I am grateful to my course teacher Dr. Hossen Asiful Mostafa who assigned us the term paper and constantly encourage us to complete it. Furthermore, he allows me to choose the above topic. Besides, I am also grateful to all the authors of the papers and owners of the reference materials because without the help from these sources it was impossible to write this paper.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, & G. Hinton, "ImageNet classification with deep convolutional neural networks", in *Proc. Advances in Neural Information Processing Systems*, pp. 1097-1105, 2012.

[2] J. Tompson, A. Jain, Y. LeCun & C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation", in *Proc. Advances in Neural Information Processing Systems*, vol. 27, pp. 1799–1807*,* 2014.

[3] T. Mikolov, A. Deoras, D. Povey, L. Burget, & J. Cernocky, "Strategies for training large scale neural network language models", in *Proc. Automatic Speech Recognition and Understanding*, pp. 196–201, 2011.

[4] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, & V. Svetnik, "Deep neural nets as a method for quantitative structure-activity relationships", J. Chem. Inf. Model, vol. 55, pp. 263–274, 2015.

[5] M. Helmstaedter, K. L. Briggman, S. C. Turaga, V. Jain, H. S. Seung, W. Denk, "Connectomic reconstruction of the inner plexiform layer in the mouse retina", *Nature*, vol. 500, pp. 168-174, 2013.

[6] M. K. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, "Deep learning of the tissue-regulated splicing code," *Bioinformatics*, vol. 30, no. 12, pp. i121–i129, Jun. 2014.

[7] H. Y. Xiong *et al.*, "The human splicing code reveals new insights into the genetic determinants of disease," *Science*, vol. 347, no. 6218, pp. 1254806–1254806, Dec. 2014.

[8] R. Collobert et al. "Natural language processing (almost) from scratch," J. Mach. Learn. Res. vol. 12, pp. 2493–2537, 2011.

[9] A. Bordes, S. Chopra, & J. Weston, "Question answering with subgraph embeddings" in *Proc. Empirical Methods in Natural Language Processing,* pp. 614-620, 2014.

[10] L. Bottou, & O. Bousquet, "The tradeoffs of large scale learning" in *Proc. Advances in Neural Information Processing Systems* vol. 20, pp. 161–168, 2007.

[11] R. O. Duda, & P. E. Hart, Pattern Classification and Scene Analysis, Wiley, 1973.

[12] Y. Bengio, P. Lamblin, D. Popovici & H. Larochelle, "Greedy layer-wise training of deep networks", in *Proc. Advances in Neural Information Processing Systems* vol. 19, pp. 153–160, 2006.

[13] P. Sermanet, K. Kavukcuoglu, S. Chintala, & Y. LeCun, “Pedestrian detection with unsupervised multi-stage feature learning”, in *Proc. International Conference on Computer Vision and Pattern Recognition*, pp. 3626-3633, 2013.
[14] Y. LeCun, et al. “Hand written digit recognition with a back-propagation network” in *Proc. Advances in Neural Information Processing Systems,* pp. 396–404, 1990.
[15] Y. LeCun, L. Bottou,Y. Bengio, & P. Haffner, “Gradient-based learning applied to document recognition” in *Proc. IEEE* vol. 86, pp. 2278–2324, 1998.
[16] D. Ciresan, U. Meier, J. Masci, & J. Schmidhuber, “Multi-column deep neural network for traffic sign classification”, *Neural Networks*, vol. 32 pp. 333–338, 2012.
[17] F. Ning et al. “Toward automatic phenotyping of developing embryos from videos”, *IEEE Trans. Image Process*. Vol. 14, pp. 1360–1371. 2005.
[18] Y. Taigman, M. Yang, M. Ranzato, & L. Wolf, “Deep face: closing the gap to human-level performance in face verification”, in *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, 2014.
[19] T. Sainath, A. R. Mohamed, B. Kingsbury, & B. Ramabhadran, “Deep convolutional neural networks for LVCSR”, in *Proc. Acoustics, Speech and Signal Processing*, pp. 8614–8618, 2013.
[20] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, “PCANet: A simple deep learning baseline for image classification?”, *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, Dec. 2015.
[21] X. Lu, Z. Lin, H. Jin, J. Yang, and J. Z. Wang, "Rating image aesthetics using deep learning," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2021–2034, Nov. 2015.
[22] S. V. Rice, F. R. Jenkins, and T. A. Nartker, “The fifth annual test of OCR accuracy,” *Information Sciences Res. Inst.*, Las Vegas, NV, TR-96-01, 1996.
[23] E. M. Newton and P. J. Phillips, “Meta-analysis of third-party evaluations of iris recognition,” *IEEE Trans. Syst., Man, Cybern. A*, vol. 39, no. 1, pp. 4–11, 200