Mid Term Assessments

Name waqar kaleem khan

Enrollment No 01-249191-013

Course Deep Learning

Teacher Dr Imran Siddqi

Submission Date 28-05-2020

Class MSDS(3-A)

Q NO 1 (i) Difference between using pre-trained models as feature extractor and fine-tunning the pre-trained model.

Ans:- The pre-trained models can be used as a seperate feature extraction program in which what we do as that input can be preprocessed by the model or portion of the model to a given an output forexample a vector of numbers for each input image. So that we can used that as ~~a~~ an input when we are training new model.

Alternately the pre-trained model or desired portion of a model can be integrated directly to a new model In this case the weights of the pre-trained model can be frozen to be not updated when training a new model otherwise the weights will be updated when training new model.

The fine tunning of a pre-trained model is that to tweak the parameters of already trained model. So the model can adopt the new task. The starting layer of the model learn very general features and as we go further to more layers the layers tend to learn more specifically about the task it is beig trained on. So for fine-tunning we want to freez or intact the initial layers and retained the later layers for our, task.

Q No 1 (ii) (i) No of employees = 500
(ii) No of picture for each employee = 10

1. we multiply 500 to 10 then overall train data set = 500 × 10 = 5000
(ii) test data piz for each employee is 2
Overall dataset = 500 × 2 = 1000

So we have 5000 images for training and 1000 images for testing
number of all images = 6000
input size of image = 100 × 100 × 3

Steps we will follow to create the system are below

(i) we will use Anaconda (jupyter Notebook)
(ii) Install the essential Libraries using "cond" or pip3 commands
(iii) After installing import that libraries to jupyter Notebook
(iv) Next step is to load the train and test data set.
(v) use numpy arrays to convert the pixel into arrays because in pyton we have one dimesional array so we use numpy.
(vi) giving shape to the data according to the model.
(vii) then split data for the model in test train if the data is not spliteed before best practices are 80% training and 20% testing means the data is prepared for model
(viii) Load the model without the clasifying part

(ix) Then we will have convolutional layer in model the data will be passed through the convolutional layer. to get the important feature in the image pixels layer which is closer to input layer will learn to detect simple feature like edges etc

(x) In the last layers add svm classifier for classifing non-linear objects because we are using images data

(xi) Train the classifier

(xii) And in the last part the output layer have 500 neuron and using softmax activation function..

QNO #2 (i) Maxpooling with filter size 3x3 padding = 1 and stride = 1

2	2	2
5	5	5
1	1	1

adding padding = 1 to above image
padding = 1 so we will add padding to the four sides of the image

0	0	0	0	0
0	2	2	2	0
0	5	5	5	0
0	1	1	1	0
0	0	0	0	0

Now appling filter for maxpooling with stride = 1
Note black color is used for filter

step = 1

0	0	0	0	0
0	2	2	2	0
0	5	5	5	0
0	1	1	1	0
0	0	0	0	0

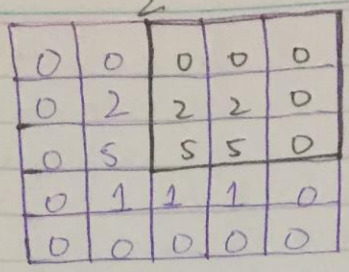
value$_1$ = 5

step = 2

0	0	0	0	0
0	2	2	2	0
0	5	5	5	0
0	1	1	1	0
0	0	0	0	0

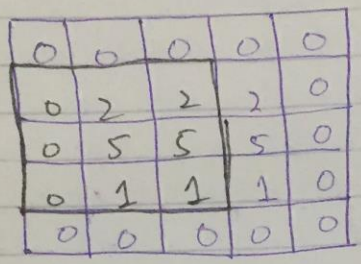
value$_2$ = 5

Step = 3



value 3 = 5

Step = 4

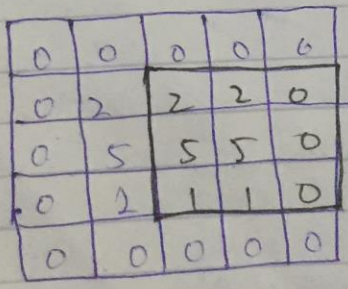


value 4 = 5

Step 5 =

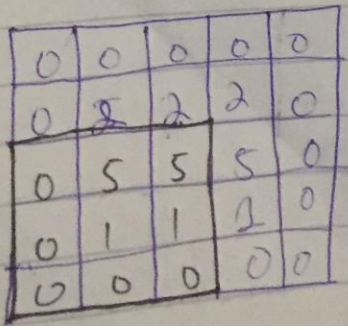
value 5 = 5

Step 6



value 6 = 5

step 7



value 7 = 5

3

Step 8

0	0	0	0	0
0	2	2	2	0
0	5	5	5	0
0	1	1	1	0
0	0	0	0	0

value 8 = 5

Step 9

0	0	0	0	0
0	2	2	2	0
0	5	5	5	0
0	1	1	1	0
0	0	0	0	0

value 9 = 5

Maxed pool values =

5	5	5
5	5	5
5	5	5

In maxpooling the parameters didnot increase It is used to preseve high values in the feature map

Q2 (ii) Briefly dissuss whether or not a CNN prescrves relative spatial realtionships between components

Ans CNN do not preserves relative spatial relation ship between components for example that we have two images of face one have every object like eye and nose etc in right place The other image objects are not in same place so to a CNN both pictures are similar because the both contain similar elements.

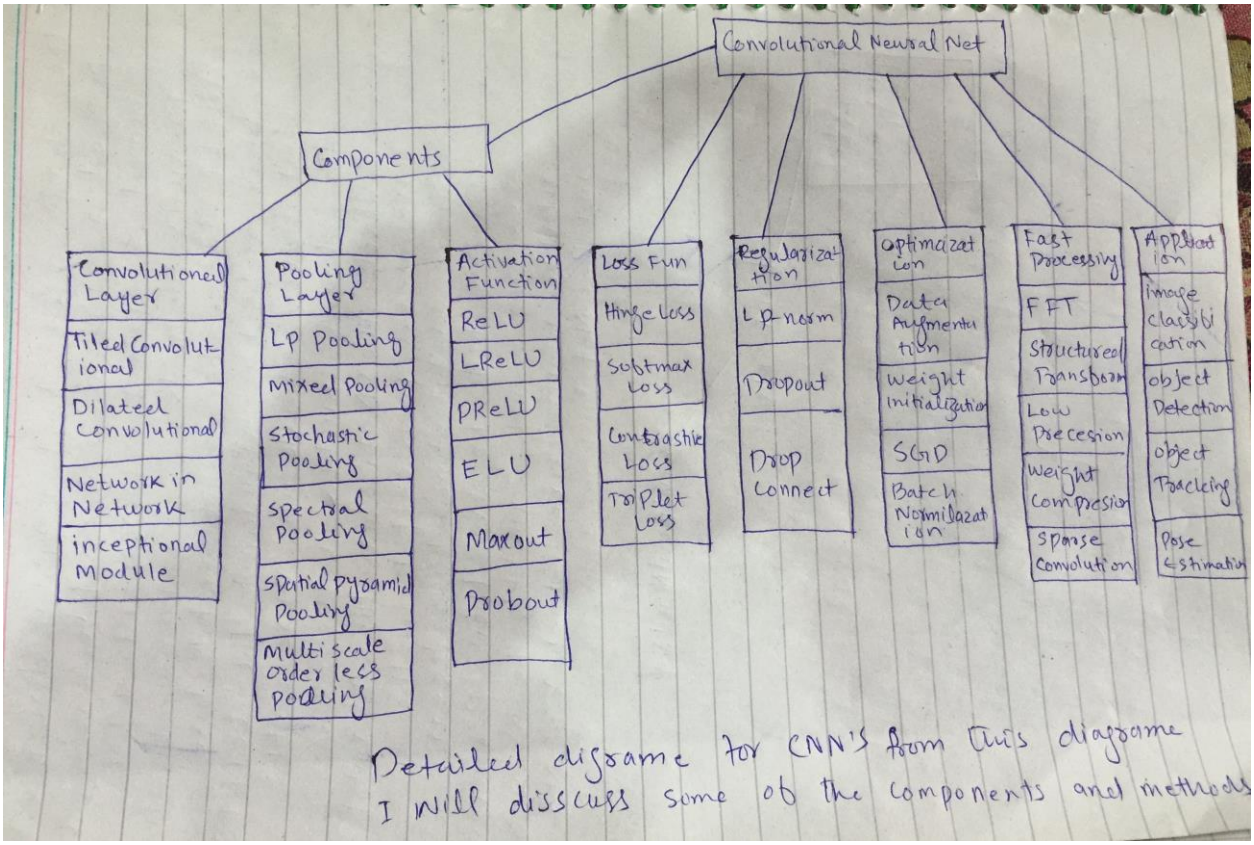
Q#3 Comprehensive analytical review on advancement in convolutional Neural Network

Abstract:- In this review we will disscuss about that how the convolutional Neural Networks have evolved and what are the popular architectures and what are the recent trends in the CNN's. If we see the history of deeplearning then we can see that in last years on the variety of problem's the deeplearning led to a very good Performance which include the area's of computer vision natural language processing (NLP) speech recognition. The CNN's among different neural network have been studied most extensively. While improvements in the graphic processor units (GPU) and the rapid growth in the annotated data the research on convolutional neural network got focused and achived state of art of results on the various tasks. In this report the CNN's improvements have been detailed using different aspects like loss function, activation function, different layers etc. and also in

last will disscuss some of the applications of the CNN's

Introduction:. In the introduction part we will disscuss how CNN'S work's what is the CNN'S architecture and history. CNN's are well known deeplearning architecture which is a memecry of humans or living visual mechanism. In 1959 two scientist named Hubel and wiesel that cells in animal visual cortex are responsible for detecting light in receptive fields. Two other computer scientist inspired by the above theory and proposed the neocughitron in 1980 which could be regard as the predecesser of CNNs. In 1990 LeCun in his research establish the moder framework of CNN. They developed a multi-layer Artificial neural network called LetNet 5. Which could be used to classify the handwritten digits but due to lack of large training dater and not have a good computational power Their model didn't perform good on the complex problems since 2006, many methods have been introduced to overcome the difficulties encountered in

The deep CNN's. Most notabley krizhevsky proposeed a classical CNN architecture and showeed significant improvements upon previous methods on the image classification task. The overall architecture of their method is similar to the LeNets but Their network was mese with the deeper structure which is calleed ALexNet. When the AlexNet have been proposeed then the other architecture have been proposeed to improve Their perbormance. From which some of the representative works are ZFnet, VGGNet and Res Net


Convolutional Neural Net
Components
Convolutional Layer
Tiled Convolutional
Dilated Convolutional
Network in Network
inceptional Module
Pooling Layer
LP Pooling
Mixed Pooling
Stochastic Pooling
Spectral Pooling
Spatial Pyramid Pooling
Multi scale order less pooling
Activation Function
ReLU
LReLU
PReLU
ELU
Maxout
Probout
Loss Fun
Hinge loss
Softmax Loss
Contrastive Loss
Triplet Loss
Regularization
Lp norm
Dropout
Drop Connect
Optimization
Data Augmentation
Weight initialization
SGD
Batch Normalization
Fast Processing
FFT
Structured Transform
Low Precesion
Weight Compresion
Sparse Convolution
Application
Image classification
object Detection
object Tracking
Pose Estimation
Detailed digrame for CNN'S from this diagrame
I will disscuss some of the components and methods

Some Basic Components of CNN's:- The basic components of the CNN's are very similar. Taking one of the most famous architecture LeNet-5 as an example. the architecture consist of three type layers which are convolutional, pooling, and fully-connected layers. The convolutional layer aims to learn feature extraction of the inputs. Convolutional layer is composed of several convolutional kernels which are used to compute different feature maps. Specifically each neuron of a feature map is connected to a region of a neuron to of a previous layer. The new feature map can be obtained by first convolving the input with a learned kernal and then applying an element wise non-linear activation function.

Mathmatically (i,j) in kth feature map of l-th layer. is calculated

$$z^l_{i,j,k} = w^{l\,T}_k x^l_{i,j} + b^l_k$$

Exploring the above formula where the w^l_k and b^l_k are the weight vector and bias term of the k-th filter of the l-th layer and $x^l_{i,j}$ is the input patch centered at location (i,j) of the l-th layer. The kernal w_k that generates the feature

map $z^l_{i,j,k}$ is shareed. Such a weight sharing mechanism has several advantages like it can reduce the model complexity and made the network easier to train
The pooling layer aims to achive shift-invariance by reducing the resolution of the feature map It is usually place between two convolutional layer. Each and every feature map of a pooling layer is connected to its corresponding feature map of preceeding convolutional layer.
matimatically we can clone the pooling as

$$y^l_{i,j,k} = Pool(a^l_{m,n,k}) \forall (m,n) \in R_{ij}$$

explaining the above matimatical form where $R_{i,j}$ is a local neighbourhood around location. The typical average operation are average pooling and max pooling In the first convolutional layer the kernal are designed to detect low features such as edges and curves. while the high layer kernal are design to learn how to encode more abstract features.
If we look to some of the CNN's architecture after several convolutional and pooling layer there may be one or more connected layers which aim to perform high level reasoning.

What in the fully connected layer happen is that all the neuroun of the previous layer are connected to the current layer for generating a global semantic information.
After the fully connected layer the last layer of the CNN's architecture are the output layer. If your model have developed for some classification then mostly softmax activation function is useed. Also SVM can be useed to combined with CNN's and can be used to solve different classification problems. Let us assume that Θ denote all the parameters of a CNN for example (weight vector, bias) by minimizing the appropriate loss function the optimum parameter for a specific task can be obtain

$$L = \frac{1}{N} \sum_{n=1}^{N} l(\Theta; y^{(n)}, o^{(n)})$$

N= desired input-out relation $(x^{(n)}, y^{(n)})$
$x^{(n)}$= is the nth input data
$y^{(n)}$= label corresponding target
$o^{(n)}$ = output of the CNN

Note:- In the first part of this review we explain how CNN's have been developed and how its working and some history of CNN

In the second part we will disscus the improvments or Advancments made to CNN's

Improvements or Advancements on CNNs:-
Since the success of AlexNet in 2012 after that there have been a lot of improvments on CNN's we will disscuss some of the major improvments but there are a lot of areas which have been improved in the CNN's some of them are Convolution layer, pooling layer, activation function, loss function, regulirization, optimization etc.

Convolutional layer:-
In basic CNN's the convolution filter is the generalized linear model (GLM) for the patch underlying of the local image. This works well for abstraction when instances of latent concepts are linearly separable For enhancing its representation ability there is some improvments which will be disscuss below

(*) Tiled Convolution:- The number of parameters can be drasticallly decrease by weight sharing mechanisum in the CNN's and it can also restrict the model to from learning other kind of invariance. Tiled CNN is a variation

for learning rotational and scale invariant features the tiled CNN's tiles and multiply features maps. with in the same layer seperate kernels can be learn and by square-root pooling the complex invarinces can be learned implicitly. The convolution operation are applied every k units wher k is the tile size to control the distance over which weights are shared. The units with in each maps will have same weights when the tiles k = 1 and to the traditional CNN the tiled CNN becomes identical.

The experiement done on the NORB and CIFAR-10 dataset which results that k=2 show the best results the tiled CNNs can Pesform better then tradicional CNNs on small time series data sets.

* Transposeed Convolution :- Transposed convolution can be seen the backword pass of a tradicional convolution. It's also known as deconvolution and fractionally stried convolution In the deconvolution we associates a single activation with multi output activation while on the other hand the tradicional convolution connects multiple input activation to a single activation. The dilation factor for the input feature map are given by the stride of deconvolution.

What deconvolution do is That first the input is unsample by the factor of stride then perform the convolution on the unsample input. Recently The deconvolution has been used for many task visualization, recognition etc

Pooling layer

Pooling layer is an important concept of CNN. It lowers the computational burden by reducing the number of connection between convolutional layers. In the pooling section we will discuss about some recent pooling methods

*) Lp pooling :-

Lp pooling is inspired by biologically complex cells. It has been theoratically analayze which suggest that Lp pooling provides better generalization than max pooling

matimatically Lp pooling can be represent

$$y_{i,j,k} = \left[\sum_{(m,n)\in R_j} (a_{m,n,k})^p\right]^{1/p}$$

$y_{i,j,k}$ => is the output of the pooling operator at location (i,j) in the k-th feature map

$a_{m,n,k}$ => is the feature value at location (m,n)

If the $p = 1$ then Lp corresponds to average pooling and when $p = \infty$ Lp reduce to max pooling.

(*) Mixed pooling :- mixed pooling is inspired by random Dropout and Dropconnect The main purpose of the max pooling method is that combination of max pooling and average pooling

Matimatically representation of max pooling fun

$$y_{i,j,k} = \lambda \max_{(m,n) \in R_{i,j}} a_{m,n,k} + (1-\lambda) \frac{1}{|R_{ij}|} \sum_{(m,n) \in R_{ij}} a_{m,n,k}$$

As we can see λ in above formula which is random value which can either be 0 or 1 which means either using average pooling or max pooling. When the model is in process of forward pass then the λ value is recorded and during back propagation it can be used.

Activation function :- A proper activation function significantly improve the performance of the CNN for a certain task In this section we will discuss one of all the most and recent used activation function.

ReLU :- Rectified Linear unit (ReLU) is one on the most notable, useable and non-saturated function

$$a_{i,j,k} = \max(z_{i,j,k}, 0)$$

Where $z_{i,j,k}$ is the input of the activation function at location (i,j) on the k-th channel. ReLU is a piecewise linear function which retain

the positive part and prune the negative part to zero. The max operation of ReLU allows to compute much faster the tanh( ) and Softmax( ). With ReLU the deep network can be trained without pre-training. At value 0 ReLU performance can be hurt at backpropagation.

Loss Function:- For a specific task it is important to choose a specific loss function. Advancement done in the loss function is (1) Hing loss (2) Softmax loss, (3) Contrastive loss (4) Triplet loss we will explain one because of 2000 word policy by examiner.

(*) **Hinge Loss:-** For the training of large margin classification Hinge loss is used like SVM The hinge loss function of multi-class SVM is defined in below

$$L_{hinge} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} \left[\max(0, 1 - \delta(y^{(i)}, j) \vec{w}^{T} \vec{x}_i) \right]^{p}$$

$\delta(y^{(i)}, j) = 1$ if $(y)^{(i)} = j$ otherwise $= -1$

if $p = 1$ Hinge loss (L_1-Loss)

if $p = 2$ Hinge loss (L_2-Loss) Which is squared Hinge

Regularization:- In the deep CNNs regularization used to reduce the overfiting. Overfiting is the unneglectable in deep CNN's. For overcoming this problem the solution's we have is lp-norm regularization, Dropout and Dropconnect we will explain one of the above advancment in the Regularization

⊛ **lp-norm Regularization:-** The term which have been penalize the model complexity the lp-norm regularization add additional term to modifies the objective of function formula

$$\epsilon(\theta, x, y) = L(\theta, x, y) + \lambda R(\theta)$$

$R(\theta) \Rightarrow$ is regularization term

$\lambda \Rightarrow$ is the regularization strength

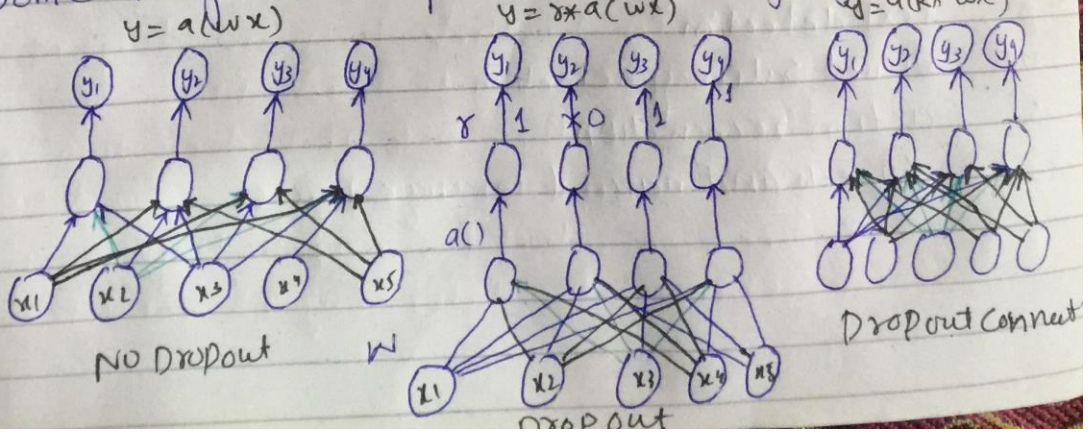
usually the lp-norm function employed as

$$R(\theta) = \sum_j \|\theta_j\|_p^p$$

if $p \geq 1$ the lp-norm is novex which means the optimization easier and renders this function

if $p=2$ then lp-norm regularization is commonly referred to as weight decay.

Optimization:- In this section we will disscuss some key tichniques for optimizing CNNs



Data Augmentation :- Deep CNNs are particularly dependent on the large quantities of training data avavalibality. So the data augmentation is the elegant solution to alleviate the relative sarcity of data compared to the number of parameters. What data augmentation do is the transform the avalible data in to the new data without altering Their nature. Some of popular augmentation methode are simple geometric transformation ~~and~~ such as mirroring, rotating, shifting. Paulin propose a greedy strategy that selects the best transformation from a set of candidate transformations. but Their strategy involves a large number of model-training which be expensive computationally.

Applications of CNNs :- In the application section we will dissuss on heigleth some recent work done using CNN's listed below are area where CNN's are applied are achive state of art Performance.

1) Image classification
2) object tracking
3) text detection
4) visual saliency detection
5) action recognition
6) speech and natural language process