Image Caption Generator Using CNN and LSTM

Swarnim Tripathi
swarnim0711@gmail.com
Galgotias University,India

Ravi Sharma
ravi.sharma@galgotiasuniversity.edu.in
Galgotias University,India

***Abstract* - For this paper, we use CNN and LSTM to become aware of the caption of the image. Image caption generation is a system that comprehends natural language processing & computer vision standards to recognize the connection of the image in English. In this research paper, we cautiously pursue a number of important concepts of photograph captioning and its familiar processes. We talk about Keras library, numpy and jupyter notebooks for the making of this paper. We also talk about flickr_dataset and CNN used for photo classification.**

***Keywords*- CNN,LSTM,image captioning, deep learning.**

INTRODUCTION

Every day we see a lot of photographs in the surroundings , on social media and in the newspapers. Humans are able to recognize photographs themselves only. We humans can pick out the photographs without their designated captions but on the other hand machines need images to get trained first then it'd generate the photograph caption automatically.

Image captioning may benefit for loads of purposes, for example supporting the visionless person using text-to-speech through real time feedback about encompassing the situation over a camera feed, improving social medical leisure with the aid of reorganizing the captions for photographs in social feed alongwith messages to speech. Facilitating kids in recognizing substances further to gaining knowledge of the language. Captions for every photograph on the world wide web can produce quicker & detailed authentic photographs exploring and indexing. Image captioning has diverse packages in numerous fields inclusive of biomedicine, commerce, internet looking and navy and many others. Social media like Instagram , Facebook etc can generate captions routinely from images

The principal goal of this research paper is to get a little bit of expertise in deep learning strategies. We use two strategies specially CNN and LSTM for image classification.

IMAGE CAPTIONING TECHNIQUES

CNN - Convolutional Neural systems are specific important neural systems that can produce information that has an information shape,for example, a 2D lattice and CNN is valuable for working with pictures. It examines pictures from left corner to the right corner and through to extricate significant highlights from the picture, and consolidates the element to characterize pictures. It can deal with interpreted, pivoted, scaled, and modified pictures. The Convolutional neural system is a profound learning calculation that takes in the info picture, allocates significance to various components/protests in the picture, and recognizes it from each other.

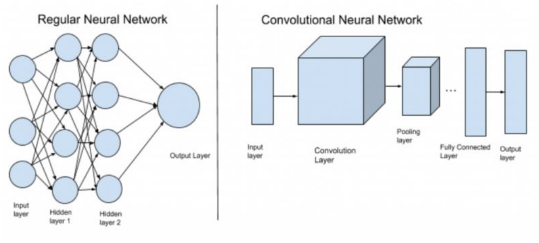


Fig.1 CNN Architecture

The required pre-handling in ConvNet negligible when compared with other order calculations. In spite of the fact that channels are hand-designed in crude strategies, with sufficient preparation, ConvNets is fit for learning these channels/highlights. The structure of the curved system is like the neuronal network design inside human mind & is inspired by the way of the organization of the visual cortex. Singular neurons react to upgrades simply in a limited district of the visible field known as open field. The assortment of such fields covers the summation of visual regions.

CNN : Architecture - A pure rustic neural network, in whatever location all neurons in a single layer merge with all of the neurons in the subsequent layer is inefficient in regards to analyzing large pictures and video. For a normal size picture with many picture elements called pixels & 3-tone colors (RGB i.e., red color,green color,blue color), the range of restriction utilizing an accepted neural system will be in the tons, & that can prompt overfitting.

To constrain effective quantities of restrictions & recognition of the neural system on significant pieces of picture, CNN utilises a 3D arrangement in which each adjustment of neurons breaks down a little area or "highlight" of picture. Rather than all neurons to skip their selections to the next neural layer, each gathering of the neurons spends significant time in distinguishing one piece of picture, such as a nose, a left ear, mouth or a leg. The last yield is a point of scope, illustrating how reasonable every one of abilities is elected as part of the class.

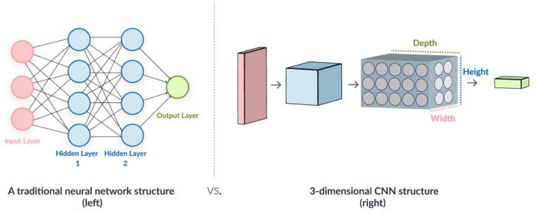
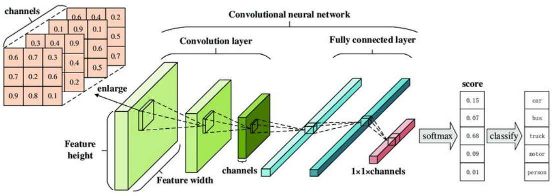

Input Layer
Hidden Layer 1
Hidden Layer 2
Output Layer
A traditional neural network structure (left)
VS.
Depth
Height
Width
3-dimensional CNN structure (right)

Fig.2 Working of CNN

How does CNN work ?

As we have discussed previously, a fully connected neural network where the input in the preceding layers is connected to every input in the following layers is convenient for the task at hand, along such lines, according to CNN, the neurons in a cell may be connected with a specific cell area before it, rather than all the neurons in a totally similar way.



This helps in reducing the complexity of the neural network and acquiring less computing power. As per new computer under standard image with the use of numbers at each pixel. When we generally compare two images we check the pixel values of each pixel. This technique only helps us to compare two identical images only but when we keep different images to compare the comparison fails. In CNN image comparison takes place piece by piece.

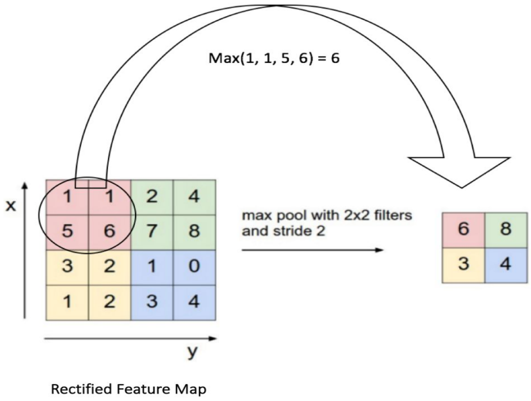


Fig.3 Feature map of CNN picture

The main reason behind using CNN algorithm is that, this is the only algorithm which takes pictures as an input and on the basis of input pictures drawing the feature map, ie.classifying each pixel on the basis on similarity and differences.The CNN classifies the pixels and a matrix is created, which is known as feature map. Feature map is a collection of similar pixels placed in a separate category. These matrices play an important role in finding the essence of the thing in the input picture.

More about CNN -

There are total 3 types of layers in CNN model-

1. Convolutional
2. Pooling
3. Fully connected

In the first layer, the input image is read through the CNN, and on that foundation a feature map is made. From that feature map , it serves as an input to the following layers, i.e for the Pooling layer. In the pooling layer, the feature map is broken down into extra simpler parts to carefully examine the context of the picture. This layer makes the feature map more dense so as to discover the most critical information about the picture.

The 1st and 2nd layers i.e Convolutional and Pooling they're practised so many times, depending on the picture as to get the densed information about the picture. The extra dense feature map is created because of these two layers. And this densed feature map is utilised by the last layer i.e Fully Connected.

This layer performs classification. It sorts the pixels with respect to similarity and differences. Classification is done upto exceptional limit so as to get the essence of the picture, help in identifying the objects , persons, things,etc.

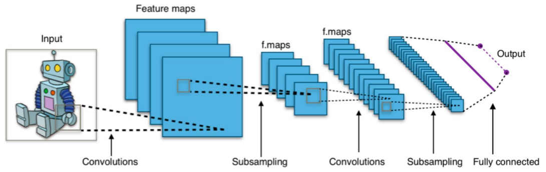


Fig.4 Layers of scanned picture

These layers help CNN to clearly locate and find features of the picture. Extraction of vital features present in the picture of fixed length inputs is transformed into fixed size outputs.

CNN techniques are very much in usage viz,

· **Computer vision**— in the area of medical sciences image analysis is done through CNNs only. Inner structure of the body is effortlessly examined with the help of this.

In mobile phones, it's been used for so many things, for instance, to find the age of the person, to unlock the phone by examining the picture from the camera.

In industries its far used for making patents or copyright of specific clicked pictures.

· **Pharmaceuticals discovery**— its been broadly used for discovering the drugs/pharmaceuticals, by analysing the chemical features and finding the best drug to cure a particular problem.

Origin of LSTM:-

LSTM was first searched by two German researchers - Sepp Hochreiter and Jurgen Schmidhuber, in 1997. LSTM stands for long short-term memory. In the Deep Learning discipline of recurrent neural networks, LSTM holds a crucial place. The special element about LSTM is that it not only stores the input data, but can also supply predictions about the subsequent datasets through its own. This LSTM network retains the stored data for a particular time period and on that basis predicts or gives the future values to the data. This is the main purpose why LSTM is used here more than that of traditional RNN.

The Problem with RNNs(Recurrent Neural Networks):-

RNNs are a part of a deep learning set of rules which are performed to deal with a number of complicated or complex computer tasks like item classification & speech recognition. RNNs are performed to address an array of activities that arise in series, with the information of every situation based completely on statistics from preceding situations.

Exquisitely, we intend to favour RNNs which are having extended collections of data & higher capabilities. This RNN can be used to carry out plenty of real life problems like inventory forecasting & reinforce speech recognition. Yet, RNNs are not used to solve real life problems & that is because of the Vanishing Gradient problem.

Vanishing Gradient Problem -

This vanishing gradient problem is the main cause which makes the working of RNNs challenging. In general, the engineering of RNNs is made such that it stores the data for some short period of time and stores some array of data. It's not possible for RNNs to remember all the data values and a long period of time. RNNs can only store some of the data for a small period of time. Thereupon, the reminiscence of RNNs is only favourable for shorter arrays of data and for short-time periods. This vanishing gradient problem becomes very prominent as compared to traditional RNNs- to solve a particular problem it adds so many time steps, which results in losing the data when we use backpropagation. With so many time steps, RNNs have to store data values of each time step, which results in storing more & more data values and that one is not feasible in the case of RNNs. And by this vanishing gradient problem is formed.

What can be done so as to solve this Vanishing Gradient problem with RNNs -

To solve this problem, we will be using Long short-term memory (LSTM) , which is a subset of RNNs. LSTM are basically constructed to overcome the problem of Vanishing Gradients. The exceptional thing about LSTM is that it can preserve the data values for lengthy interval of time and hence can solve the vanishing gradient problem.

LSTMs are constructed in such a manner that they always contain errors. And due to these errors LSTM keeps studying the data values over several time steps. Because of studying data values again & again, it makes studying backpropagation easy over time & layers.

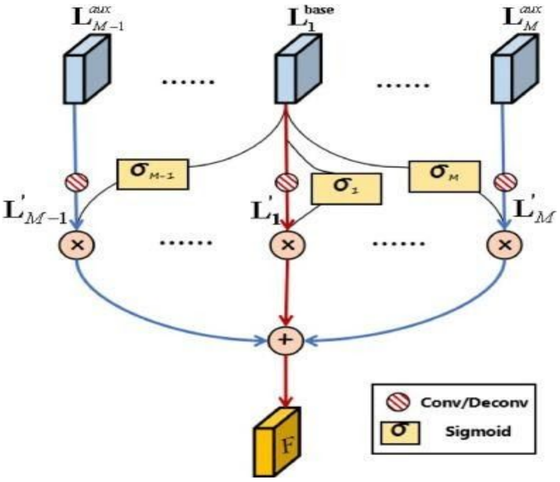


Fig.5 Gates in LSTM

Ideally, as per the diagram above, LSTM uses several gates to store the data and after that it processes the data and sends the result to the final gate. When we talk about RNNs, they used to pass the data to the final gate without any processing. From these gates in LSTM, the whole network can shape the data in many forms, including storing the data and reviewing the data from the gates. The gates in the LSTM are independently skillful to make judgement concerning the facts & the data. Moreover these gates are able to make judgements on their own by opening or closing the gates.

The understanding of LSTM gates to hold the data for a period of time offers benefit to the LSTM over the RNNs.

<u>Architecture Of LSTM</u> :-

The architecture of LSTM is very simple, it consists of 3 major gates, which store the data for a longer period of time and help in solving the difficulties which RNNs couldn't solve.

The 3 major gates of the LSTM covers are :

· **Forget gate** — the main work of the forget gate is to filter the data, i.e. to delete all that data which is not needed in the future to solve a particular task. This gate is responsible for the overall performance of the LSTM, it optimizes the data.

· **Input gate** — the starting of LSTM starts from this gate, i.e. input gate. This gate takes input from the user and supplies the input data to other gates.

· **Output gate** — This gate is responsible for showcasing the desired result in a proper manner.

<u>Uses of Long Short-Term Memory Networks :-</u>

LSTMs are profoundly and mostly used for variety deep learning duties that largely encompasses forecasting of the data depending upon the preceding data. The 2 remarkable illustrations cover text prediction and stock market prediction.

Text Prediction - The LSTM is very used in predicting the texts. The long term memory, understanding of LSTM makes it capable enough to predict the next words in the sentences. This is the result of the LSTM network in predicting the next words by its own. The LSTM first stores the data, the feel of the words, the styling of the words, the use of the words in a particular situation,etc and on that basis predicts the next words. The stored data, i.e. input data is further used for future use.

The best illustration can be given of text prediction is a Chatbot, that is widely utilized by the eCommerce websites and mobile applications.

Stock market Prediction - In the stock market also, LSTM stores the data or the trends in which the market behaves at a particular time, at a particular instant and on that account predicts the next variations and trends of the market. It's a problematic task to predict the variation in the stock market because market variations are very challenging to predict and forecast. The LSTM model has to be trained in such a manner that it gives the correct values to the users. For that, a lot of data has to be stored for a lot of time, it can take days also.

<u>More about LSTM</u> -

LSTMs are basically a part of RNNs, which are having capacity to hold more data values as compared to RNNs. LSTMs are widely in use today in every field. The simplest diagram of LSTM is shown below. It consists of 3 major gates viz, Forget gate, input gate, output gate. These gates are having capacity to store the data and give out the desired output. Whenever talked about LSTM network the three gates always comes up. The below diagram shows the simplest architecture of LSTM :

Fig.6 Working of LSTM

Image Caption Generation Model:-

In order to prepare an image caption generation model, we will be summing up the two different architectures. It is further called as CNN-LSTM model. So, in this we will be using these two architectures to get the caption for the input pictures.

· CNN - it's been used to extract the important features from the input picture. To do this, we have taken a pre-trained model for our consideration named Xception.

· LSTM - its been used to store the data or the features from the CNN model and further process it and to support in the generation of a good caption for the picture.

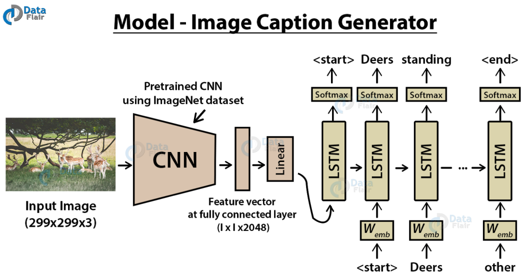


Fig.7 CNN-LSTM model

Project File Architecture :-

For our research purpose, we have downloaded the data set which consists of following files :

· **Flickr8k_Datasets –** This file contains all the pictures for which we have to first train our model. It contains 8091 images.

· **Flickr8k_texts –** This folder contains text files & pre-formed captions for the pictures.

The following files are set up for making this system to run by us to check the working of the CNN-LSTM model..

· **Model –** This folder will contain all the trained models which are at first trained. This would be one time process to train the model.

· **Description.txt –** This is the file which will contain the picture names & their related captions later preprocessing.

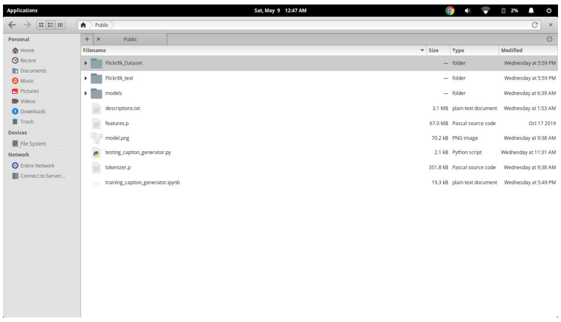
· **Feature.p –** This file binds the picture and their related captions that are extracted from the Xception, which is a pre-trained CNN model.

· **Tokenizers.p –** This file contains an expression which we call tokens , and these tokens are generalised with the index value.

· **Models.png –** Diagrammatic representation of extension of the CNN-LSTM model.

· **Testing_captions_generator.py –** This is the Python file which is used in generating the captions of the pictures.

· **Training_captions_generator.ipynb –** This is basically a Jupyter notebook, which is in short a web based application. We use this to train our model & on that basis achieving captions to our input pictures.



Filename	Size	Type	Modified
Flickr8k_Dataset	—	folder	Wednesday at 5:59 PM
Flickr8k_text	—	folder	Wednesday at 5:59 PM
models	—	folder	Wednesday at 6:39 AM
descriptions.txt	3.1 MB	plain text document	Wednesday at 1:53 AM
features.p	67.0 MB	Pascal source code	Oct 17 2019
model.png	70.2 kB	PNG image	Wednesday at 9:38 AM
testing_caption_generator.py	2.1 kB	Python script	Wednesday at 11:31 AM
tokenizer.p	351.8 kB	Pascal source code	Wednesday at 9:38 AM
training_caption_generator.ipynb	19.3 kB	plain text document	Wednesday at 5:49 PM

Fig.8 Project file structure

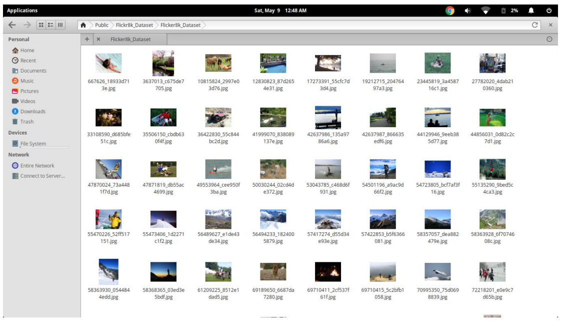


Fig.9 Flickr_Dataset

Conclusion

The CNN-LSTM model was built on the idea of generating the captions for the input pictures. This model can be used for a variety of applications. In this, we studied about the CNN model, RNN models, LSTM models, and in the end we validated that the model is generating captions for the input pictures.

References

[1] Abhaya Agarwal and Alon Lavie. 2008. Meteor, m-bleu and m-ter: Evaluation metrics for high-correlation with human rankings of machine translation output. In Proceedings of the ThirdWorkshop on Statistical Machine Translation. Association for Computational Linguistics, 115–118.

[2] Ahmet Aker and Robert Gaizauskas. 2010. Generating image descriptions using dependency relational patterns. In Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 1250–1258.

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In European Conference on Computer Vision. Springer, 382–398.

[4] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2017. Bottom-up and top-down attention for image captioning and vqa. arXiv preprint arXiv:1707.07998 (2017).

[5] Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. 2018. Convolutional image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 5561–5570.

[6] Lisa Anne Hendricks, Subhashini Venugopalan, Marcus Rohrbach, Raymond Mooney, Kate Saenko, Trevor Darrell, Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, et al. 2016. Deep compositional captioning: Describing novel object categories without paired training data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In International Conference on Learning Representations (ICLR).

[8] Shuang Bai and Shan An. 2018. A Survey on Automatic Image Caption Generation. Neurocomputing. ACM Computing Surveys, Vol. 0, No. 0, Article 0. Acceptance Date: October 2018. 0:30 Hossain et al.

[9] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, Vol. 29. 65–72.