NOVATEUR PUBLICATIONS
INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY
[IJIERT] ISSN: 2394-3696 Website: ijiert.org
VOLUME 8, ISSUE 7, July. -2021

IMAGE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS BY DIFFERENT ACTIVATION FUNCTIONS

Nayoumi Mytholla
Gitam University , Hyderabad, India
m.nayoumi@gmail.com

Vodala Chakshu
Gitam University, Hyderabad, India
chakshuvodala8@gmail.com

ABSTRACT

With the continuous development of deep learning, convolution neural network with its excellent recognition performance obtains a series of major breakthrough results in target detection, image recognition and other fields. An improved ReLu segmentation correction Activate function is proposed, by improving the traditional convolution neural network, adding the local response normalization layer, and using the maximum stacking and so on. Based on the deep learning consepts, the activation function is used to construct the modified convolution neural network structure model, using the Boat analysis data set as the neural network input for the model training and evaluation. We analyze effects of different neuron activation function on the neural network convergence speed and the accuracy of image classification. The experimental results show that the boat image is classified based on different activation functions used.

Keywords: Image classification, Convolotion neural network, Accuracy.

INTRODUCTION

A neural network is a very powerful machine learning mechanism which basically mimics how a human brain learns.The brain receives the stimulus from the outside world, does the processing on the input, and then generates the output. As the task gets complicated, multiple neurons form a complex network, passing information among themselves. An Artificial Neural Network tries to mimic a similar behavior. The network you see below is a neural network made of interconnected neurons. Each neuron is characterized by its weight, bias and activation function. The input is fed to the input layer, the neurons perform a linear transformation on this input using the weights and biases.

$$x = (\text{weight} * \text{input}) + \text{bias}$$

Finally, the output from the activation function moves to the next hidden layer and the same process is repeated. This forward movement of information is known as the forward propagation.What if the output generated is far away from the actual value,Using the output from the forward propagation, error is calculated. Based on this error value, the weights and biases of the neurons are updated. This process is known as back-propagation.

Imagine a neural network without the activation functions. In that case, every neuron will only be performing a linear transformation on the inputs using the weights and biases. Although linear transformations make the neural network simpler, but this network would be less powerful and will not be able to learn the complex patterns from the data. Thus we use a non linear transformation to the inputs of the neuron and this non-linearity in the network is introduced by an activation function. Artificial neural networks are inspired from the biological neurons within the human body which activate under certain circumstances resulting in a related action performed by the body in response. Artificial neural nets consist of various layers of interconnected artificial neurons powered by activation functions which help in switching them ON/OFF. Like traditional machine algorithms, here too, there are certain values that neural nets learn in the training phase. Briefly, each neuron receives a multiplied version of inputs and random weights which is then added with static bias value (unique to each neuron layer), this is then passed to an appropriate activation function which decides the final value to be given out of the neuron. There are various activation functions available as per the nature of input values. Once the output is generated from the final neural net layer, loss function (input vs output)is calculated and backpropagation is performed where the weights are adjusted to make the loss minimum. Finding optimal values of weights is what the overall operation is focusing around.

Weights are numeric values which are multiplied with inputs. In backpropagation, they are modified to reduce the loss. In simple words, weights are machine learnt values from Neural Networks. They self-adjust depending on the difference between predicted outputs vs training inputs. Activation Function is a mathematical formula which helps the neuron to switch ON/OFF.

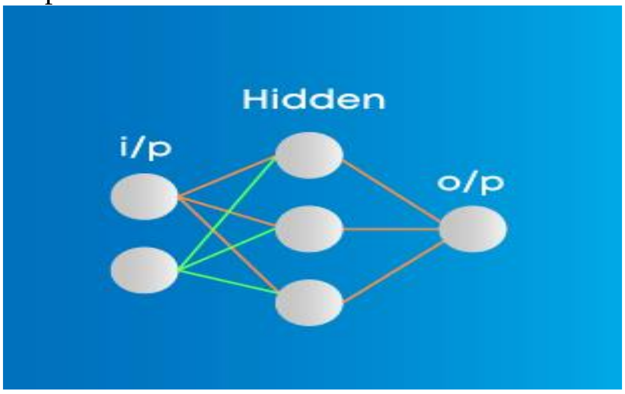

Figure 1.Input and output layer

- Input layer represents dimensions of the input vector.
- Hidden layer represents the intermediary nodes that divide the input space into regions with (soft) boundaries. It takes in a set of weighted input and produces output through an activation function.
- Output layer represents the output of the neural network.

Types of Neural Networks

Feedforward Neural Network – Artificial Neuron:

This neural network is one of the simplest forms of ANN, where the data or the input travels in one direction. The data passes through the input nodes and exit on the output nodes. This neural network may or may not have the hidden layers. In simple words, it has a front propagated wave and no backpropagation by using a classifying activation function usually.

Radial basis function Neural Network:

Radial basic functions consider the distance of a point with respect to the center. RBF functions have two layers, first where the features are combined with the Radial Basis Function in the inner layer and then the output of these features are taken into consideration while computing the same output in the next time-step which is basically a memory.

Recurrent Neural Network(RNN) – Long Short Term Memory:

The Recurrent Neural Network works on the principle of saving the output of a layer and feeding this back to the input to help in predicting the outcome of the layer.Here, the first layer is formed similar to the feed forward neural network with the product of the sum of the weights and the features. The recurrent neural network process starts once this is computed, this means that from one time step to the next each neuron will remember some information it had in the previous time-step.

Convolutional Neural Network:

Convolutional neural networks are similar to feed forward neural networks, where the neurons have learnable weights and biases. Its application has been in signal and image processing which takes over OpenCV in the field of computer vision.Below is a representation of a ConvNet, in this neural network, the input features are taken in batch-wise like a filter. This will help the network to remember the images in parts and can compute the operations.

Modular Neural Network:

Modular Neural Networks have a collection of different networks working independently and contributing towards the output. Each neural network has a set of inputs that are unique compared to other networks constructing and performing sub-tasks. These networks do not interact or signal each other in accomplishing the tasks.The advantage of a modular neural network is that it breakdowns a large computational process.

LITERATURE SURVEY

The various types of CNN, designed and implemented successfully in various fields of image processing and object recognition. Its better if you have an idea of ConvolutionalNeural Network.You probably have heard of ImageNet. It is a large organized visual image database used by researchers and developers to train their models. Now, they host an annual competition named ImageNet Large Scale Visual Recognition Challenge (ILSVRC) a competition related to object detection and image classification on a large scale. Generally, the top performers of this competition are able to set a benchmark in the field of object classification.

LeNet:

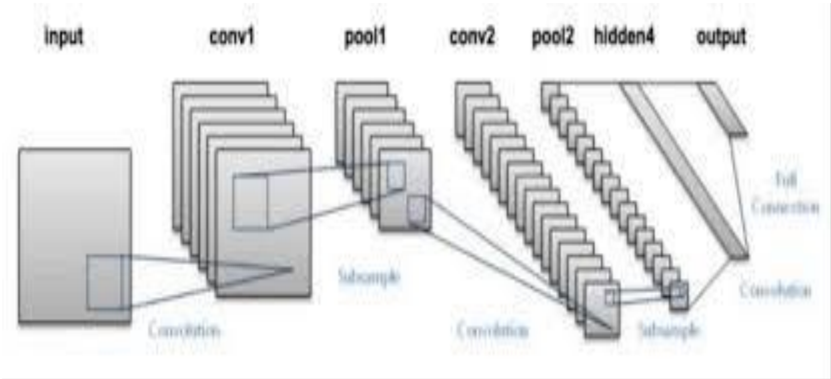

Figure 2.LeNet architecture

No discussion of the CNN architectures can begin without this. A ground-breaking algorithm that was the first of its kind and capability, in-terms-of object classification. Originally trained to classify hand written digits from 0–9, of the MNIST Dataset**.** It comprises of 7 layers, all made of trainable parameters. It takes in a 32 X 32 pixel image, which was comparatively large in size w.r.t the images present in the data sets on which the network was trained.

AlexNet:

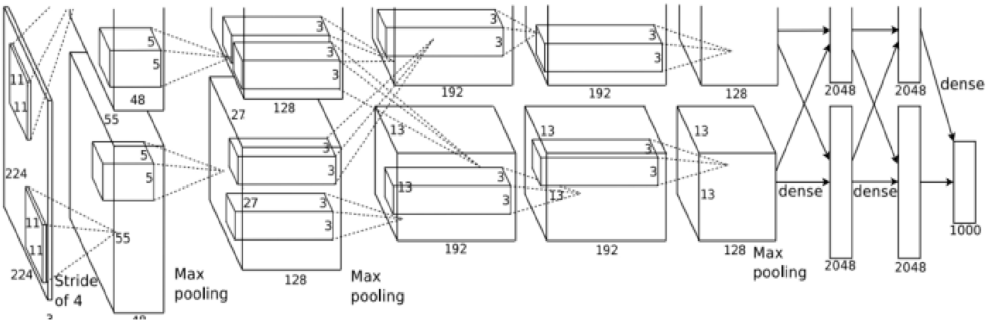

Figure 3. Alexnet architecture

AlexNet**,** the winner of the ImageNet ILSVRC-2012 competition, was designed by Alex Krizhevsky, Ilya Sutskever and Geoffery E. Hinton**.** It was able to reduce the top-5 error rate to 15.3 % compared to the error rate of the runners-up of that competition which attained an error rate of 26.2%. The network is similar to the LeNet Architecture, but has a large no. of filters compared to the original LeNet.

VGGNet 16:

This particular network architecture was the runners up of the ILSVRC-2014competition, designed by Simonyan and Zisserman**.** It was ale to achieve a top-5 error rate of 5.1%**.** Though it might look complicated with a whole bunch of parameters to be taken care of, it is actually very simple. Developers prefer it highly, when it comes to feature extraction because of the simple pattern that it follows.

ResNets:
Probably after AlexNet, the most ground-breaking development in the field of CNN architecture development happened with ResNet or Residual Networks**.** This is based on the idea of "skip-connections" and implements heavy batch-normalization, that help it in training over thousands of layers effectively, without degrading the performance in the long run.The problem rose with the training of deeper networks.

METHODOLOGY

Activation function:
Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a neuron.

Variants of activation function: -
1). Linear Function: -

- **Equation:** Linear function has the equation similar to as of a straight line i.e. **y = ax**
- No matter how many layers we have, if all are linear in nature, the final activation function of last layer is nothing but just a linear function of the input of first layer.
- **Range:** -inf to +inf
- **Uses:** Linear activation function is used at just one place i.e. output layer.
- **Issues:** If we will differentiate linear function to bring non-linearity, result will no more depend on input "x" and function will become constant, it won't introduce any ground-breaking behavior to our algorithm.
-

2). Sigmoid Function:-

- It is a function which is plotted as 'S' shaped graph.
- **Equation :** $A = 1/(1 + e^{-x})$
- **Nature :** Non-linear. Notice that X values lies between -2 to 2, Y values are very steep. This means, small changes in x would also bring about large changes in the value of Y.
- **Value Range :** 0 to 1
- **Uses :** Usually used in output layer of a binary classification, where result is either 0 or 1, as value for sigmoid function lies between 0 and 1 only so, result can be predicted easily to be 1 if value is greater than 0.5 and ***0*** otherwise.

3). Tanh Function :- The activation that works almost always better than sigmoid function is Tanh function also knows as Tangent Hyperbolic function**.** It's actually mathematically shifted version of the sigmoid function. Both are similar and can be derived from each other.

4). RELU :- Stands for Rectified linear unit. It is the most widely used activation function. Chiefly implemented in hidden layers of Neural network.

- **Equation :-** A(x) = max(0,x). It gives an output x if x is positive and 0 otherwise.
- **Value Range :-** [0, inf)
- **Nature :-** non-linear, which means we can easily backpropagate the errors and have multiple layers of neurons being activated by the ReLU function.
- **Uses :-** ReLu is less computationally expensive than tanh and sigmoid because it involves simpler mathematical operations. At a time only a few neurons are activated making the network sparse making it efficient and easy for computation.

5). Softmax Function: - The softmax function is also a type of sigmoid function but is handy when we are trying to handle classification problems.

- **Nature:-** non-linear
- **Uses:-** Usually used when trying to handle multiple classes. The softmax function would squeeze the outputs for each class between 0 and 1 and would also divide by the sum of the outputs.
- **Output: -** The softmax function is ideally used in the output layer of the classifier where we are actually trying to attain the probabilities to define the class of each input.

Image classification and modeling based on deep convolution neural network

The following is a model for image classification based on deep convolution neural networks.
Input: Input is a collection of *N* images; each image label is one of the *K* classification tags. This set is called the training set.
Learning: The task of this step is to use the training set to learn exactly what each class looks like. This step is generally called a training classifier or learning a model.
Evaluation: The classifier is used to predict the classification labels of images it has not seen and to evaluate the quality of the classifiers. We compare the labels predicted by the classifier with the real labels of the image. There is no doubt that the classification labels predicted by the classifier are consistent with the true classification labels of the image, which is a good thing, and the more such cases, the better.
The convolution layer is the most important part of a convolutional neural network. The input of each node in the convolution layer is only a small part of the upper layer of the neural network. The convolution layer analyses every smaller part of the neural network in depth, and as far as possible to get a higher degree of feature abstraction. The pooling layer does not change the depth of the three-dimensional matrix in the neural network, but it will reduce the size of the matrix, that is, reduce the number of nodes in the next layer, so as to reduce the parameters of the whole neural network and decrease the training time. After multiple rounds of convolutions and pooled layers, the information in the image has been abstracted into a higher information content, and the full connection layer is used to complete the classification task. The fully connected layer performs the combination matching and classification by modifying the nonlinear activation function, mainly used for classification problems, through the Soft max layer, you can get the sample belongs to different types of probability.
ReLu is easy to calculate, simple to achieve and has fast convergence speed, so it can effectively alleviate the gradient disappearance problems, and provide a certain sparse characteristic for the neural networks after training， more in line with the nature of biological neuron activation. The Softplus function is an approximate smooth representation of the ReLu function, with unilateral suppression properties, and wider excitation boundary, but it does not have better sparsity. Although the Softsign function is similar to the hyperbolic tangent Tanh, the synchronization is more robust due to its smoother asymptotic line, the relatively slow and soft saturation. The activation value using the Softsign function is uniformly distributed in a large number of nonlinear but the gradient flow of good area, has better fault tolerant ability. On the basis of traditional convolution neural network, this paper does data enhancement, adds local response normalization, uses overlapping maximum pooling and and other improvements. As the ReLu activation function can effectively alleviate the disappearance of the gradient and sparseness, combining with Softsign activation function which has the characteristics of high degree of non - linearization and good fault - tolerant ability, we propose an improved ReLu segmentation correction activation function. Each single layer convolution neural network consists of three basic stages: convolution feature extraction, nonlinear activation and down-sampling. Convolution neural network is a trainable multi-layer network structure composed of multiple stacks of single-layer convolution neural networks.

RESULT

The Boat analysis dataset has 9 ships associated with it.Using the code we have found out 1162 files belonging to 9 classes.Using 232 files for validation. Now displaying the images belong to 9 different class are given below.

Figure 4.Different ships

Then normalizing the input layer using preprocessing steps we are able to apply different activation functions at the input layer using convolution neural network . The sequential model is used to predict the metrics like accuracy and validation accuracy , loss by fitting the given model.After successful epochs the metrics like accuracy is determined. The accuracy of 95% is obtained for the boat analysis and at the we are able to precict image most likely belongs to cruise_ship with a 67.65 percent confidence.

```
epochs=10
history = model.fit(train_ds,validation_data=val_ds,epochs=epochs)

Epoch 1/10
30/30 [==============================] - 162s 5s/step - loss: 1.9893 - accuracy: 0.3226 - val_loss: 1.7728 - val_accuracy: 0.3879
Epoch 2/10
30/30 [==============================] - 34s 1s/step - loss: 1.6420 - accuracy: 0.4258 - val_loss: 1.7727 - val_accuracy: 0.3491
Epoch 3/10
30/30 [==============================] - 34s 1s/step - loss: 1.4514 - accuracy: 0.5086 - val_loss: 1.6218 - val_accuracy: 0.4612
Epoch 4/10
30/30 [==============================] - 35s 1s/step - loss: 1.1838 - accuracy: 0.6118 - val_loss: 1.5641 - val_accuracy: 0.5129
Epoch 5/10
30/30 [==============================] - 35s 1s/step - loss: 1.0140 - accuracy: 0.6656 - val_loss: 1.6392 - val_accuracy: 0.4957
Epoch 6/10
30/30 [==============================] - 35s 1s/step - loss: 0.8782 - accuracy: 0.7054 - val_loss: 2.1717 - val_accuracy: 0.4828
Epoch 7/10
30/30 [==============================] - 35s 1s/step - loss: 0.8078 - accuracy: 0.7419 - val_loss: 1.7864 - val_accuracy: 0.5043
Epoch 8/10
30/30 [==============================] - 35s 1s/step - loss: 0.4343 - accuracy: 0.8591 - val_loss: 1.9597 - val_accuracy: 0.5259
Epoch 9/10
30/30 [==============================] - 35s 1s/step - loss: 0.2519 - accuracy: 0.9237 - val_loss: 2.3933 - val_accuracy: 0.4526
Epoch 10/10
30/30 [==============================] - 35s 1s/step - loss: 0.1700 - accuracy: 0.9516 - val_loss: 2.4221 - val_accuracy: 0.5302
```

Figure 5. Accuracy and Loss

CONCLUSION

The activation function is an important part of the convolution neural network, which can map the nonlinear features of the data, so that the convolution neural network has enough ability to capture the complex pattern. On the basis of the traditional convolution neural network, this paper enhances data, adds the local response normalization layer, and using the maximum pooling and so on. Besides the problem of insufficient expression of the Relu function, And the softsign activation function is nonlinear and the improved fault tolerance, an improved ReLu segmentation correction activation function is proposed. Based on the Google deep learning platform TensorFlow, this paper uses the activation function to construct the modified convolution neural network structure model. The Boat data set is used as the neural network input to train and evaluate the model. The effect of different neuron activation functions on network convergence speed and image recognition accuracy is compared and analyzed through experiments. The experimental results show that the proposed improved activation function in image classification results in excellent, faster convergence speed, effectively alleviate the problem of the gradient diffusion model, and improves the image recognition accuracy of neural network.

REFERENCES

1) Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]. International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1097-1105.
2) Lecun Y, Boser B, Denker J S, et al. Backpropagation Applied to Handwritten Zip Code Recognition[J]. Neural Computation, 2014, 1(4):541- 551.
3) Mass A L, Hannun A Y, Ng A Y. Rectifier Nonlinearities Imporove Neural Network AcousticModels[C/OL]. 2016001.https://web.stanford.edu/~awni/papers/relu/hybrid_icml2013_final.pdf.
4) Iliev A, Kyurkchiev N, Markov S. On the approximation of the step function by some sigmoid functions[J]. Mathematics & Computers in Simulation, 2017, 133:223-234.
5) Ali Hamidoğlu. On general form of the Tanh method and its application to nonlinear partial differential equations[J]. Numerical Algebra, Control\s&\soptimization:naco, 2017, 6(2):175-181.
6) Senior A, Lei X. Fine context, low-rank, softplus deep neural networks for mobile speech recognition[C]// IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2014:7644-7648.
7) Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C]// International Conference on International Conference on Machine Learning. Omnipress, 2010:807-814.