IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Resnet18 Model With Sequential Layer For Computing Accuracy On Image Classification Dataset

[1]Allena Venkata Sai Abhishek, [2]Dr. Venkateswara Rao Gurrala, [3]Dr. Laxman Sahoo

[1] Research Scholar, [2] Professor, [3]Senior Professor

[1]Department of Computer Science and Engineering,

[1]GITAM University, Visakhapatnam, Andhra Pradesh, India

Abstract: This residual network has been a broad domain of research in deep learning. Many complex architectures are based upon residual networks. Residual networks are efficient due to skip connections. This paper highlights the addition of a sequential layer to the traditional RESNET 18 model for computing the accuracy of an Image classification dataset. The classification datasets such as Intel Scene dataset, CIFAR10 dataset, etc. These datasets consist of images belonging to various classes. In classification, we assign the pictures to their respective categories. The addition of a sequential layer gives the accuracy in the range of 0 to 1, which helps find the accuracy of prediction for the test set.

***Index Terms* – RESNET18, Sequential Layer, Dropout Layer, Adam Optimizer, NLL Loss Function, ReLU, Classification.**

I. INTRODUCTION

The neural networks have become deeper and deeper in recent times, ranging in over a hundred layers. RESNET18 [11] [13] [14] is an artificial neural network that is also called Residual Network. The residual network framework helps ease the network's training which is more profound and deeper than the usual artificial neural network. Artificial Intelligence is putting human intelligence in machines to mimic human actions. Machine learning is a subset of Artificial intelligence that helps the machine learns with data and help in making decisions. It is achieved with minimal human intervention. Deep learning is the subpart of machine learning inspired by the human brain's structure. There are many frameworks that RESNET can use such as Tensor flow [6], Keras [8], Torch etc. The brief about artificial intelligence is given in Fig. 1

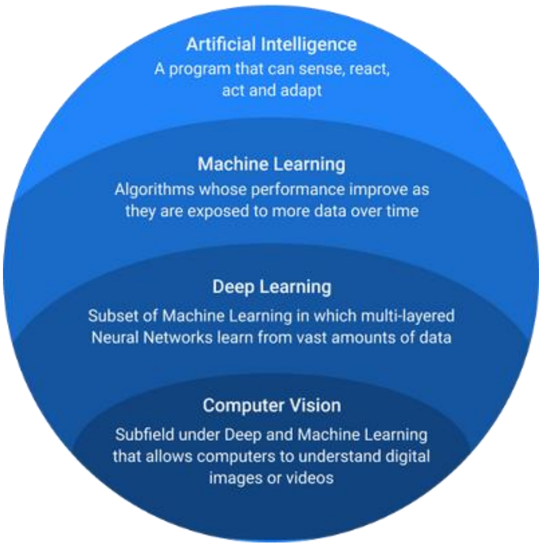

Figure 1: Deep Learning

II. BAFCKGROUND WORK

A deep neural network can represent a very complex function. An enormous obstacle for very deep networks is vanishing gradients in which the gradient of the network goes quickly to zero, subsequently making gradient descent slower and slower.

In gradient descent, we backpropagate from the last layer to the first layer of the network, and the weight matrix is multiplied at every step. If the gradient is slight, the gradient reduces exponentially to zero due to an immense number of multiplications. The deeper networks have a degradation problem in which the model's performance degrades with an increase in several layers.

In the research paper, Microsoft introduced the deep residual network that attempts to solve the degradation problem using this framework. A residual network is an approach in which a shortcut is added. The shortcut is called a skip connection.

The skip connection helps inflow the information easily from one layer to the next to the next layer, which means the data is bypassed with the standard Convolutional Neural Network flow from one layer to the next to the next layer.

The idea is to use the network to fit the residual mapping instead of learning from the underlying mapping.

$H(x) = F(x) + x$ as the mapping that occurs initially for $H(x)$, there is a usage of, $F(x) = H(x) - x$

The RESNET has many variants –

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 2: RESNET18 variants

The each variant consists of different number of layers with 18, 34, 50,101,152 layers named –

- RESNET18
- RESNET34
- RESNET50
- RESNET101
- RESNET152

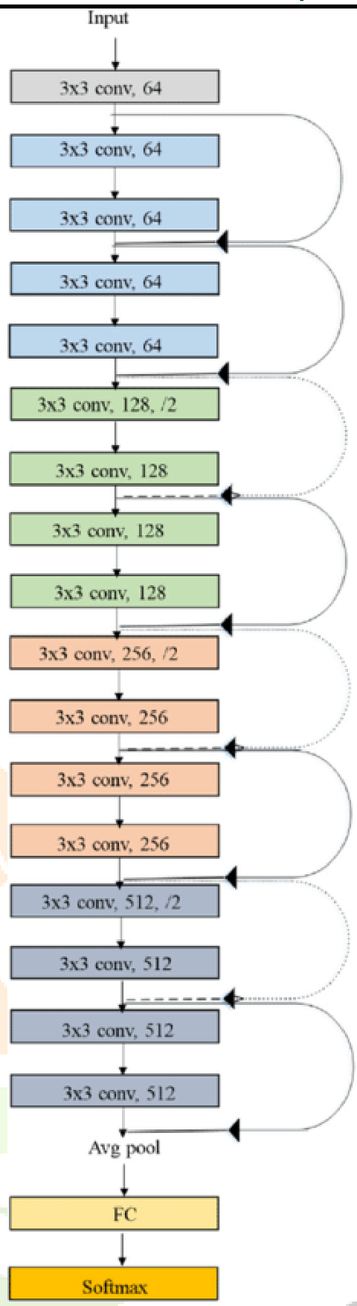

Figure 3: RESNET18

The ReLU [9] is added and applied after adding the skip connection. The residual block consists of a skip connection added after every two-weight layer.

III. Research Methodology

RESNET18 has 18 layers with a 7x7 kernel as 1st layer. It has four layers of ConvNets that are identical. Each layer consists of two residual blocks. Each block consists of two weight layers with a skip connection connected to the output of the second weight layer with a ReLU. If the result is equal to the input of the ConvNet layer, then the identity connection is used. But, if the input is not similar to the output, then a convolutional pooling is done on the skip connection.

The input size taken by the RESNET18 is (224, 224, 3), which is done by applying augmentation using AugStatic library [1] [2] in pre-processing step [7]. In (224, 224, 3) where 224 is the width and height. 3 is the RBG channel. The output is an FC layer that gives input to the sequential layer [3] [10].

In the methodology proposed, there is an additional sequential layer at the end of the last layer of the RESNET18, as mentioned in fig 6. The input will be passed to linear (512,512), whose output is fed to the first ReLU activation function. Then a dropout (0.2) layer is being used, followed by linear (512, 2). Finally, it is passed through a LogSoftmax to get the logarithm of the probabilities. The model uses Pytorch framework [4]. It is shown in Fig. 4 and Fig. 5.

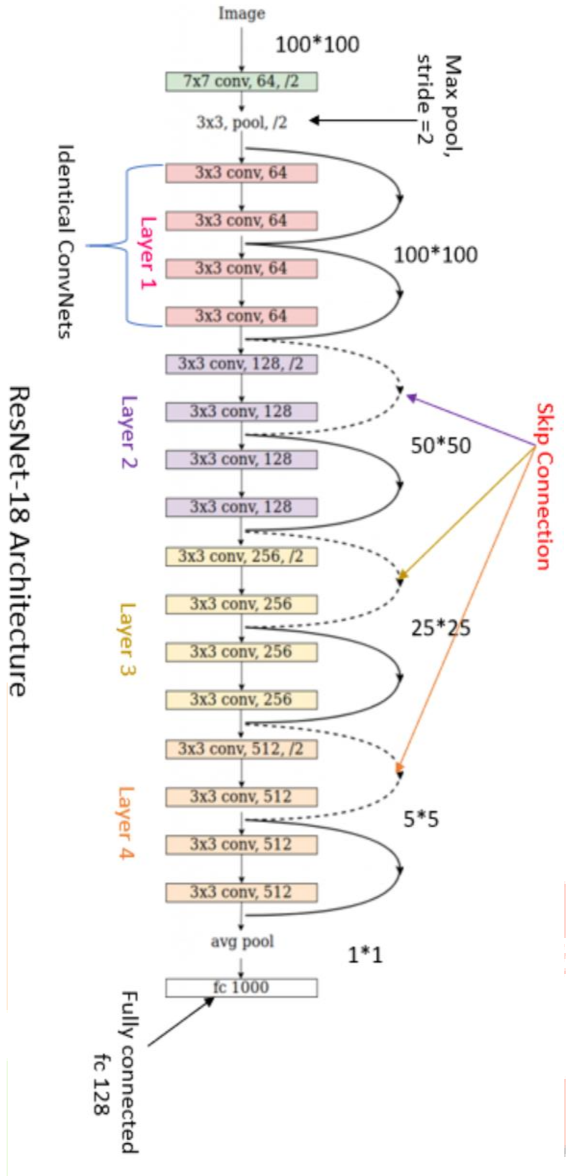

Figure 4: ResNet-18 Base Architecture

```
(fc): Sequential(
  (0): Linear(in_features=512, out_features=512, bias=True)
  (1): ReLU()
  (2): Dropout(p=0.2, inplace=False)
  (3): Linear(in_features=512, out_features=2, bias=True)
  (4): LogSoftmax(dim=1)
```

Figure 5: ResNet-18 Additional Sequential Architecture

Adam optimizer is being used. The advantage of adam optimizer with some extensions to stochastic gradient descent is the Adaptive Gradient Algorithm (AdaGrad) which improves the performance on problems with small gradients and maintains a parameter learning rate. Another advantage of adam optimizer is Root Mean Square Propagation (RMSProp), which is based upon the averaging of the gradients for the 4elearning rates of the weight parameter. The algorithm of Adam optimizer is shown in Fig. 6.

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
$m_0 \leftarrow 0$ (Initialize 1st moment vector)
$v_0 \leftarrow 0$ (Initialize 2nd moment vector)
$t \leftarrow 0$ (Initialize timestep)
while θ_t not converged **do**
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
$\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
$\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
end while
return θ_t (Resulting parameters)

Figure 6: Adam Optimizer

NLL Loss Function, i.e., the negative log-likelihood loss, is used. The 1D Tensor [5] assigning weight to each class should be the optional argument weight. This function is particularly useful in the case of an unbalanced training set. An example of the negative log likelihood loss function is shown in Fig 7.

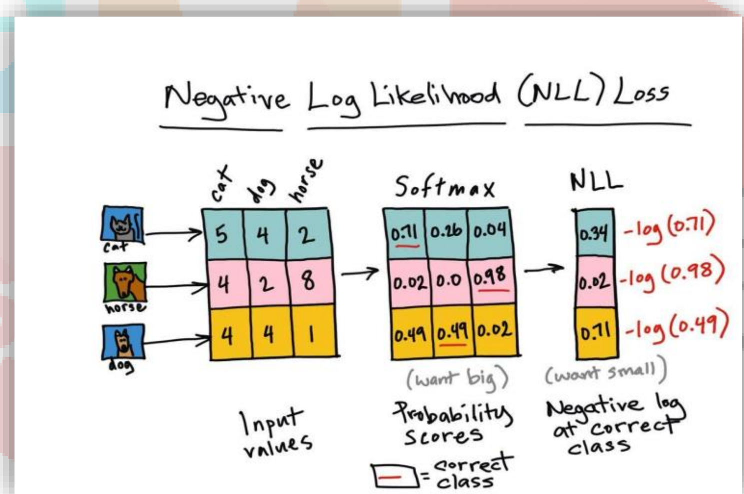

Figure 7: Negative Log Likelihood Loss function

IV. Results and Discussions

The accuracies of the RESNET18 architecture with an additional sequential layer consisting of a linear (512,512) layer, whose output is fed to the first ReLU activation function followed by a dropout (0.2) and a linear (512, 2). Finally, it is passed through a LogSoftmax to get the logarithm of the probabilities, which were found using the negative log-likelihood loss function. The network can be used to find the accuracy of the image dataset, which can be used for classification [12] or detection.

V. Conclusion And Future Scope

There are millions of neural networks, out of which the residual network is the efficient and complex neural network that is effective in finding the accuracies on various image datasets. The sequential layer helps acquire the accuracies in the range of 0 to 1 using linear, dropout layers, and ReLU functions followed by LogSoftmax. The complexity can be increased by increasing the number of residual block layers, which helps the model learn better.

REFERENCES

[1] Allena Venkata Sai Abhishek, DR. Venkateswara Rao Gurrala. May 2022. "AUGMENTED BALANCED IMAGE DATASET GENERATOR USING AUGSTATIC LIBRARY", IJRAR - International Journal of Research and Analytical Reviews (IJRAR), E-ISSN 2348-1269, P- ISSN 2349-5138, Vol.9, Issue 2

[2] Allena Venkata Sai Abhishek, Dr. Venkateswara Rao Gurrala. May 2022. "AUGSTATIC - A LIGHT-WEIGHT IMAGE AUGMENTATION LIBRARY ", International Journal of Emerging Technologies and Innovative Research (www.jetir.org | UGC and issn Approved), ISSN:2349-5162, Vol.9, Issue 5

[3] Allena Venkata Sai Abhishek, Sonali Kotni, 2021, Detectron2 Object Detection & Manipulating Images using Cartoonization, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10, Issue 08 (August 2021)

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, Soumith Chintala. 2019. “PyTorch: An Imperative Style, High-Performance Deep Learning Library” 5arXiv:1912.01703 [cs.LG]

[5] Dimitrios Koutsoukos, Supun Nakandala, Konstantinos Karanasos, Karla Saur, Gustavo Alonso , Matteo Interlandi “Tensors: An abstraction for general data processing.”

[6] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, Xiaoqiang Zheng (2016) “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems” arXiv:1603.04467 [cs.DC]

[7] Sujith Jayaprakash Balamurugan E. 2015. A Comprehensive Survey on Data Preprocessing Methods in Web Usage Mining.

[8] Ketkar, Nikhil. (2017). “Introduction to Keras.” 10.1007/978-1-4842-2766-4_7.

[9] Abien Fred Agarap (2019) “Deep Learning using Rectified Linear Units (ReLU)” arXiv:1803.08375 [cs.NE]

[10] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, Ole Winther (2016) “Sequential Neural Models with Stochastic Layers” arXiv:1605.07571 [stat.ML]

[11] Elena Limonova, Daniil Alfonso, Dmitry Nikolaev, Vladimir V. Arlazarov (2020) “ResNet-like Architecture with Low Hardware Requirements” arXiv:2009.07190v2 [cs.CV]

[12] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton (2012) “ImageNet Classification with Deep Convolutional Neural Networks “ NeurIPS Proceedings

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun (2015) “Deep Residual Learning for Image Recognition” arXiv:1512.03385v1 [cs.CV]

[14] Allena Venkata Sai Abhishek and Venkateswara Rao Gurrala,Improving Model Performance and Removing the Class Imbalance Problem Using Augmentation, International Journal of Advanced Research in Engineering and Technology (IJARET). 13(5), 2022, pp. 14-22. https://iaeme.com/Home/issue/IJARET?Volume=13&Issue=5