Abstract

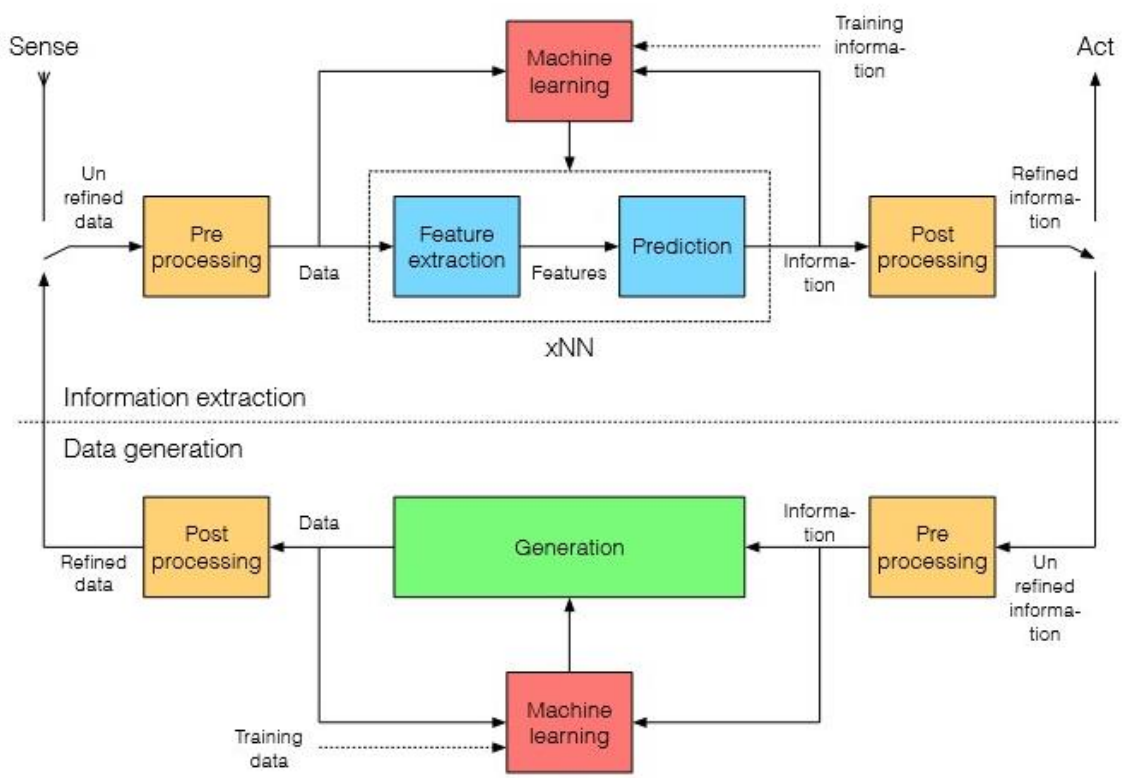
We know that deep learning has played a vital role in image recognition. By intuition, deeper the layer better the accuracy. But observation indicates that deeper networks face many problems as the depth of network increases, prominent ones being that they are more difficult to train due to difficulty of propagating error sensitivity through so many layers during back propagation (Vanishing Gradient Problem). As we increase the depth or no. of layers, we reach a saturation point where instead of improvement in accuracy it starts degrading.

This problem was alleviated to some extent using normalization techniques like batch normalization, which normalizes the activations of every single layer so that they have zero mean and unit variance for every batch of input dataset in training. Recently it has been found that this problem can be addressed more effectively using a residual network.

Residual networks help in having substantially deeper network with less complexity and is easier to train. This has been proved by comprehensive evidence in the Microsoft research paper [1], which we are implementing in this project. We are using CIFAR-10 image dataset, which is 32 x 32 image dataset, instead of 256 x 256 image dataset (which has been used in the paper [1]) due to computational constraint.

Methodology

Deep learning

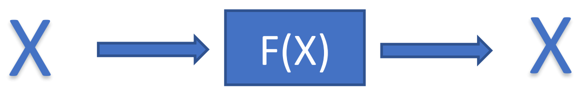


Deep learning is based on multi stack layer architecture where every layer is connected to its subsequent layers and output of one layer is input to the next layer. This is done to extract weak features from 3D images and strong features from those weak features which are then passed to neural network to get the target classes. The hypothesis is that by training a network we are trying to predict the unreferenced function to which if give images as input then we get appropriate class label in output.

Deep learning models can extract more excellent features than its shallower counterpart. Training of these networks involve computation of error at the end of network. This error sensitivity is then passed to whole network using back propagation after each cycle thereby updating the weights at each layer. The problem arises as the depth of network increases because, the error gradient computed at the start of the network becomes so close to zero that the network does not train and instead of improving it becomes worse at one point.

Identity Mapping

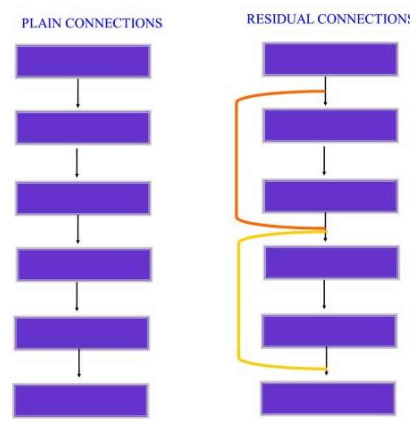
Identity mapping are the functions in which input is equal to its output.



We can convert any shallower network to deeper network by adding layers of identity mapping. A degradation is expected as the network grows deeper, but this network didn't show any signs of accuracy degradation compared to the original shallower network despite the addition of identity mappings.

Residual Functions

To solve the gradient vanishing problem associated with deep network, the authors introduced residual connection into the network. Residual connections are simply short connections between layers as shown above.



It is noteworthy that while the authors originally considered residual connections as being important for depth, future work has proven that residual networks can improve the performance of both shallow and deep neural networks.

Short connections

In short connections, a few intermediate layers are directly connected to auxiliary classifier as shown in the figure 2 of research paper [1].

The role of short connections is to ensure that adding more layers will not cause any performance degradations.

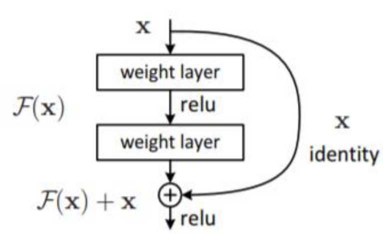
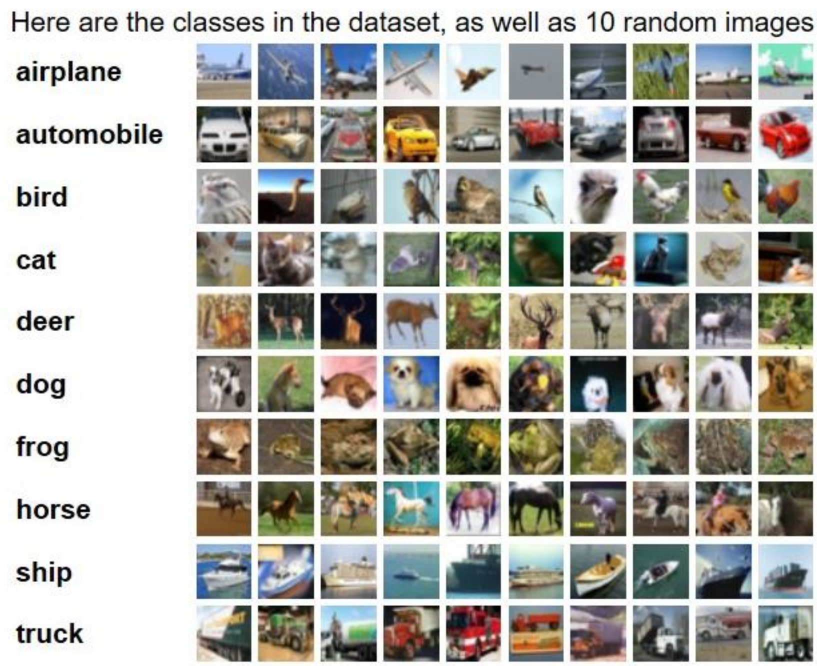


Figure 2. Residual learning: a building block.

Descriptions

We are using CIFAR-10 image dataset(https://www.cs.toronto.edu/~kriz/cifar.html), which is 32 x 32 x 3 image dataset. The dataset consists of 60000 color images (RGB) comprise of 10 classes, with 6000 images per class. We have divided this dataset into 50000 for training and 10000 for testing.



The classes are completely mutually exclusive. There is no overlap between automobiles and trucks. "Automobile" includes sedans, SUVs, things of that sort. "Truck" includes only big trucks. Neither includes pickup trucks.

We have implemented both sequential and residual models for the above dataset to observe the difference between result of both the models.

Architecture of the both the network is described in the below diagrams.

In the residual network, if you denote the desired underlying mapping as H(x), we let the stacked nonlinear layers fit another mapping of $F(x) := H(x)-x$. The original mapping is recast into F(x)+x. We hypothesize that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping. To the extreme, if an identity mapping were optimal, it would be easier to push the residual to zero than to fit an identity mapping by a stack of nonlinear layers.

The formulation of F(x)+x can be realized by feed-forward neural networks with “shortcut connections” (Fig. 2). Shortcut connections are those skipping one or more layers. In our case, the shortcut connections simply perform identity mapping, and their outputs are added to the outputs of the stacked layers (Fig. 2). Identity shortcut connections add neither extra parameter nor computational complexity. The entire network can still be trained end-to-end by stochastic gradient descent with backpropagation, and can be easily implemented without modifying the solvers.

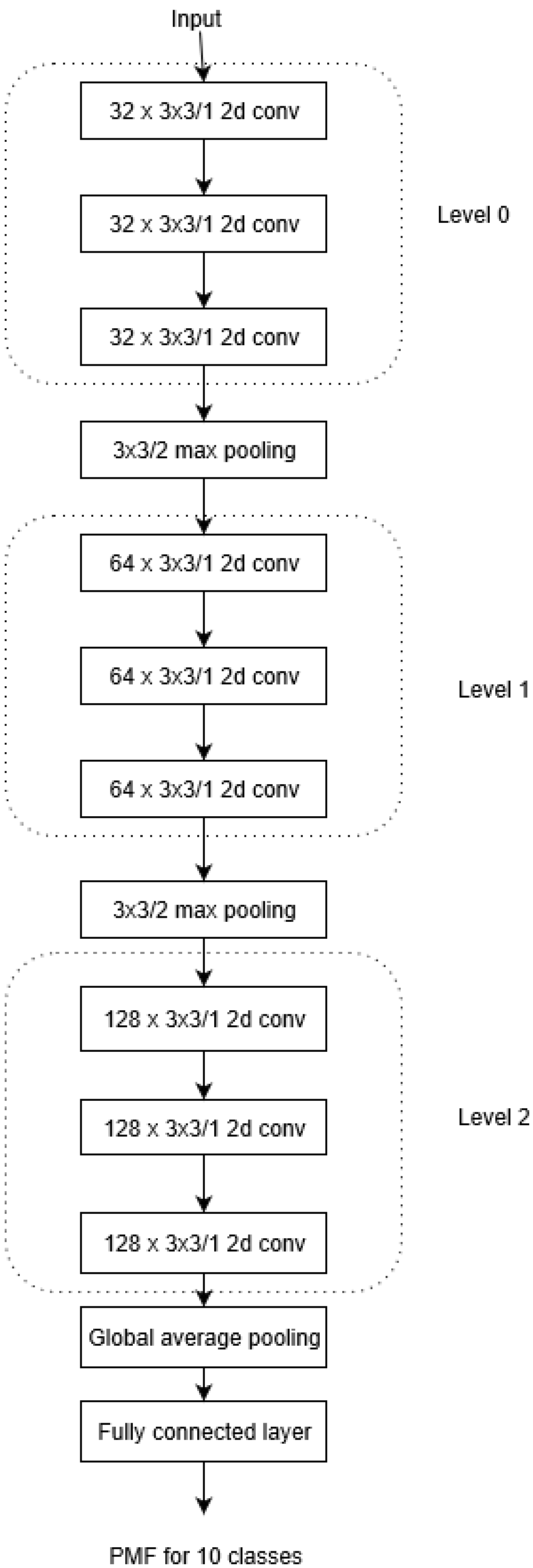


Figure 1: Sequential Network

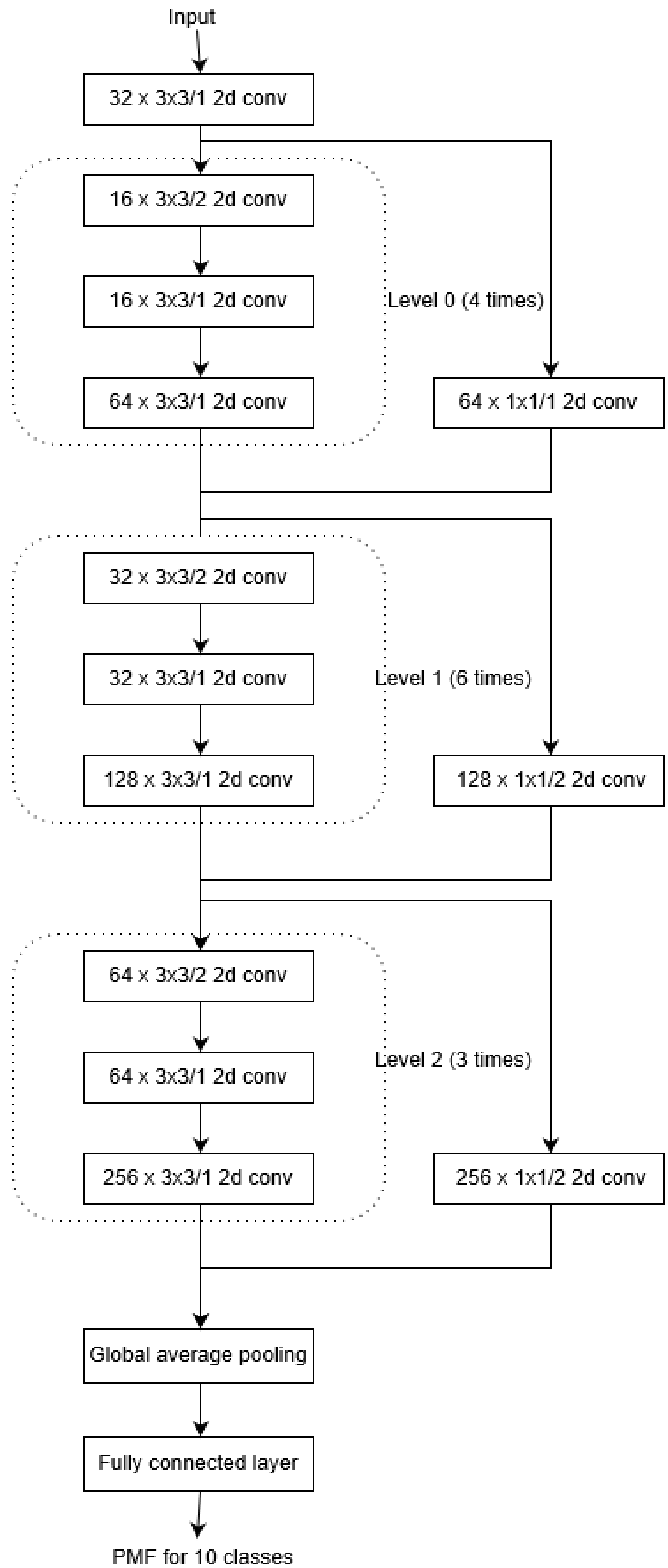


Figure 2: Residual mapping

Tools/Libraries Used

- TensorFlow
- Keras
- Google colab
- Numpy
- Matplotlib

Result

After running above described model for 20 epochs, the observation are as follows: -

Sequential Network	Residual Network
Epoch 0: top 1 accuracy on the test set is 52.39 %	Epoch 0: top 1 accuracy on the test set is 61.35 %
Epoch 1: top 1 accuracy on the test set is 64.25 %	Epoch 1: top 1 accuracy on the test set is 64.06 %
Epoch 2: top 1 accuracy on the test set is 69.11 %	Epoch 2: top 1 accuracy on the test set is 71.74 %
Epoch 3: top 1 accuracy on the test set is 73.53 %	Epoch 3: top 1 accuracy on the test set is 74.48 %
Epoch 4: top 1 accuracy on the test set is 75.45 %	Epoch 4: top 1 accuracy on the test set is 76.08 %
Epoch 5: top 1 accuracy on the test set is 75.95 %	Epoch 5: top 1 accuracy on the test set is 76.50 %
Epoch 6: top 1 accuracy on the test set is 77.90 %	Epoch 6: top 1 accuracy on the test set is 80.26 %
Epoch 7: top 1 accuracy on the test set is 76.77 %	Epoch 7: top 1 accuracy on the test set is 80.98 %
Epoch 8: top 1 accuracy on the test set is 78.19 %	Epoch 8: top 1 accuracy on the test set is 80.86 %
Epoch 9: top 1 accuracy on the test set is 78.66 %	Epoch 9: top 1 accuracy on the test set is 80.58 %
Epoch 10: top 1 accuracy on the test set is 79.20 %	Epoch 10: top 1 accuracy on the test set is 82.39 %
Epoch 11: top 1 accuracy on the test set is 80.17 %	Epoch 11: top 1 accuracy on the test set is 83.08 %
Epoch 12: top 1 accuracy on the test set is 80.69 %	Epoch 12: top 1 accuracy on the test set is 84.67 %
Epoch 13: top 1 accuracy on the test set is 79.62 %	Epoch 13: top 1 accuracy on the test set is 83.98 %
Epoch 14: top 1 accuracy on the test set is 80.90 %	Epoch 14: top 1 accuracy on the test set is 83.68 %
Epoch 15: top 1 accuracy on the test set is 81.23 %	Epoch 15: top 1 accuracy on the test set is 85.64 %
Epoch 16: top 1 accuracy on the test set is 80.29 %	Epoch 16: top 1 accuracy on the test set is 85.29 %
Epoch 17: top 1 accuracy on the test set is 81.46 %	Epoch 17: top 1 accuracy on the test set is 84.86 %
Epoch 18: top 1 accuracy on the test set is 80.05 %	Epoch 18: top 1 accuracy on the test set is 83.72 %
Epoch 19: top 1 accuracy on the test set is 81.13 %	Epoch 19: top 1 accuracy on the test set is 85.69 %

Conclusion

Residual networks are very effective in countering the vanishing gradient problem and also improving the model performance by preventing the degradation on increasing the depth of the network.

References

1. https://arxiv.org/pdf/1512.03385.pdf
2. https://arxiv.org/pdf/1603.05027.pdf
3. https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624
4. https://medium.com/deepreview/review-of-deep-residual-learning-for-image-recognition-a92955acf3aa