Object detection using Residual Networks

Rahul Agarwal
22B3961

21 January 2024

1 What architecture did we used so far?

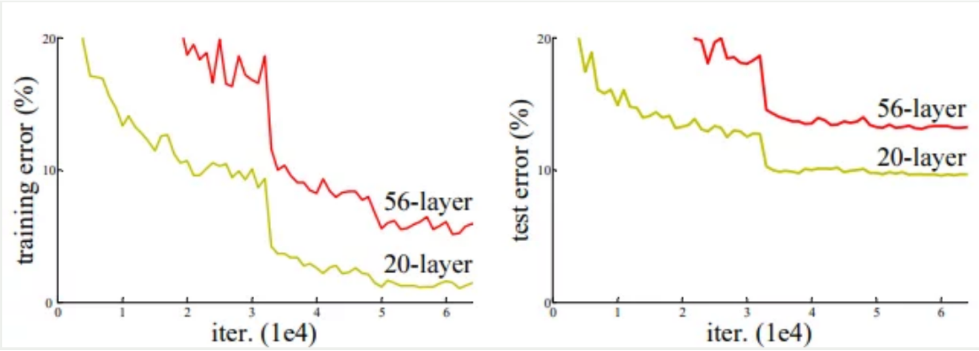
In the earlier days (before 2015 era), we used CNNs which were a huge success. They optimized the model on how our eyes classifies an image by breaking an image into multiple sub-parts and then go on with the classification process.
It involved primarily of:

- **Convolution layer -** We 'convolve' our image with filters that basically define what features are we looking for and what would be the dimensions of the image to proceed with.
- **Pooling layer -** This step is to preserve important feature information as well as reducing the dimensions of the image so as to preserve only those pixels that are crucial for the calculations.
- **Fully connected layer -** These are the layers we used in a normal neural network. Here the subparts of the image club together to help in the complete figure recognition.

The CNNs were very great compared to the conventional neural network, but the issue of vanishing and exploding gradient came into the picture.

2 Vanishing gradient and exploding gradient

If we thoroughly look into the CNNs, let's say we have a model A with 20 layers and a model B with 56 layers. Essentially, if model A has a test error of some quantity **e**, then model B (56 layers) should not have a test error greater than e, because fundamentally more layers mean more complex calculations and better trained model. But the results came out to be different:



As we can see in the figure, in model B, both, the test error and the training error, are greater than that in model A.
The reason of this could be understood this way:

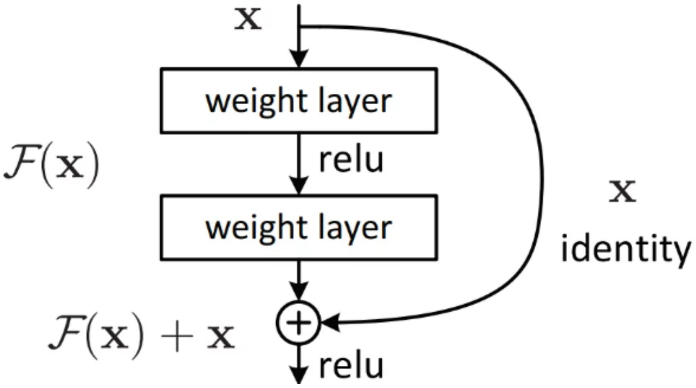
- As the network goes deep in CNNs, the problem of vanishing gradient arises, which means that the change in gradient becomes so low as it travels through backpropagation into so many layers
- Thus the model essentially does not causes much change after some time and leaves behind a huge training as well as test error.
- **Another issue** : While CNNs are much better for classification and calculations, they don't intrinsically know how to do unity mapping 'easily'. i.e. if we think about it, in model B, if the model trains the first 20 layers as it is in model A, and then do unity mapping for the rest 36 layers, the error should be atleast equal, and not greater, to that in model A.
- CNNs take a lot of calculations and parameters just even to a basic unity mapping task, and thus deep neural networks fail to perform better.

3 What are residual networks, and how did they solve the problem

The basic building blocks of a residual network are the so-called 'residual blocks'. These residual blocks provide us with what is called skip connections. These skip connections essentially skip over 'some' of the convolution layers and provide an alternate path for the gradient to travel back. They solve the problem in these 2 possible manners:

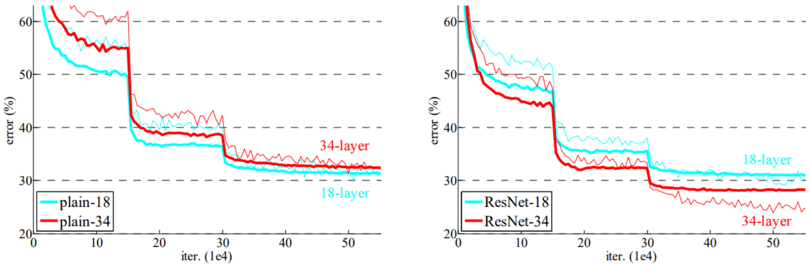
1. **Solving vanishing gradient**: The skip connections provide an alternate path for the gradient to backpropagate and hence the 'precious' gradient information is not vanished through the initial layers as it did in the

CNNs. This helps a deep model to train accurately and minimize error.


x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

2. **Unity mapping**:

   - As discussed earlier, unity mapping is bit of a task for CNNs, as they start with 0 and tweak the parameters until the mapping is achieved.
   - This issue resolves with ResNets as with the skip connections, the output is initialized by the input of the layer, and any required changes are done afterwards.
   - What this essentially means is that unity mapping is the fundamental mapping and any other changes are done around it. And thus deep networks now don't have the issue of error greater than less deep networks.



4 Advantages and Disadvantages

Every new breakthrough comes with bunch of advantages and disadvantages of their utilization over other potential methods available:

1. **Advantages**:
    - **Improved accuracy:** As we intuitively predicted earlier, a deep model involves more complexity and will give better results. ResNets provides more accuracy than conventional deep neural networks.
    - **Faster convergence:** The ResNets ensure faster convergence due the associated skip connections that provide for better gradient flow during backpropagation and skipping the vanishing gradients.
    - **Better generalization:** ResNets are found to be generalizing the model more efficiently and working pretty accurately on unseen data.
2. **Disadvantages**:
    - **Complexity:** With the deep networks, we get the model to be more and more complex and thus requiring greater computational requirements.
    - **Overfitting:** If the training dataset is small, then the model can overfit due to its complexity, and therefore proper regularization is necessary.
    - **Interpretability:** Since the model is very complex, it becomes difficult to define what it is doing at each step and thus becomes less interpretable.

Though these disadvantages are model specific, we must be precautious of them.

5 Conclusion and citations

Thus we conclude that using ResNets over conventional deep networks brings better accuracy and faster training, solves the problem of vanishing gradient and makes unity mapping much easier.

The above document was a collection of information from various sources listed below:

1. ***Convolutional Neural Networks playlis**t, DeepLearningAI*
2. ***Deep Residual Learning for Image Recognition**, Microsoft research*
3. ***ResNet: Residual Neural Networks – easily explained!**, Data Base Camp*

Following is the repository that contains the python notebook files I achieved and their ascending accuracy that Y achieved with the model in the course of this project:

- ***Image-detection-using-ResNets***