Deep Learning: Image Classification Using Convolutional Neural Networks

Chau Tran, Troy Eggertsen, Nabil Hentabli

Department of Mathematics, University of California, Santa Barbara

Overview

- We use Convolutional Neural Networks (CNNs) to identify what object is in a given picture.
- By training this network architecture on a large dataset consisting of pre-labeled images, the network learns to identify objects through pattern recognition.
- Training and testing is done using CIFAR-10.

Neural Networks

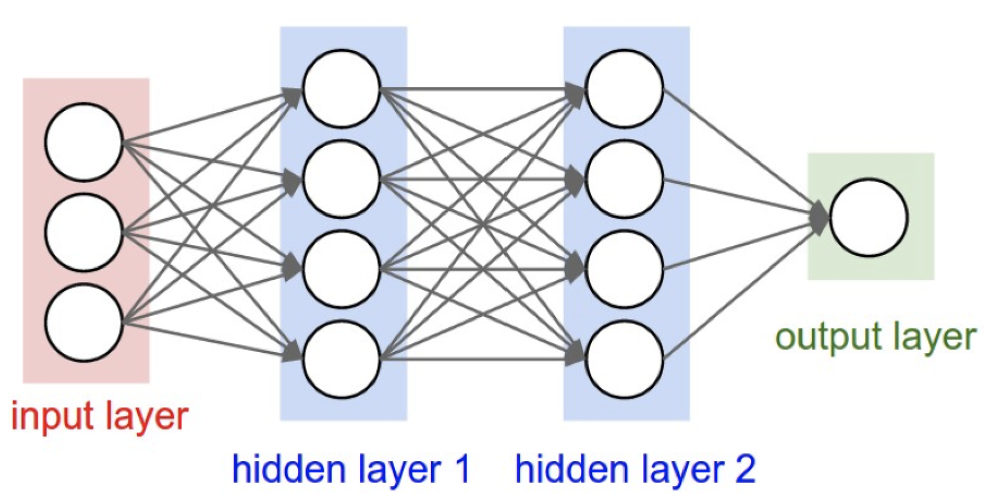

Fig. 1: Fully Connected Neural Network Architecture[5]

- Neural Networks are composed of a number of layers:
  - The input layer, in our case a three-dimensional tensor encoding an input image.
  - An arbitrarily large number of hidden layers that operate repeatedly on the outputs of the previous. layers via linear matrix multiplications and non-linear activation functions.
  - The output layer; for image classification problems, we use the softmax function to produce class confidence scores.

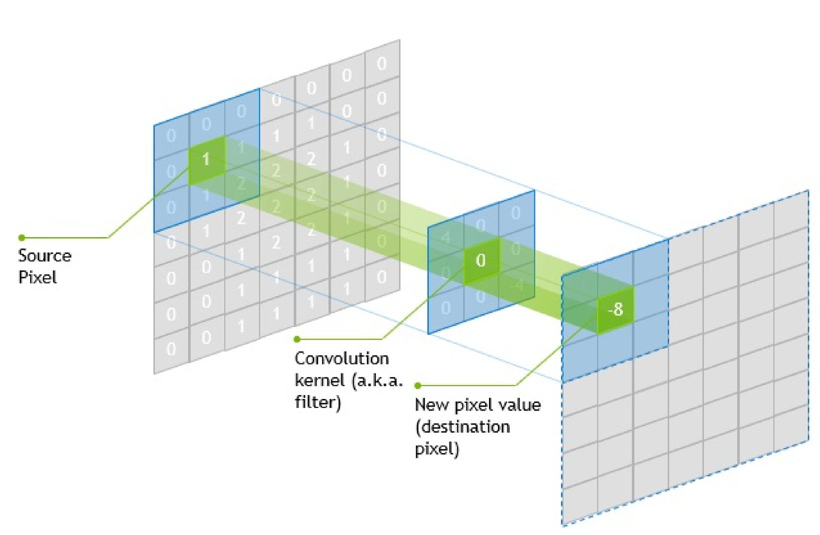

Fig. 2: Convolution [3]

- A CNN differs from standard Neural Network by containing a number of convolution layers before fully connected ones.
  - Fully connected layers are layers where every node in one layer is connected to all nodes in the next.
  - Convolution layers are better suited than standard layers for identifying information in spatial data; they do this by changing the dimensions of matrices as they go through the model.

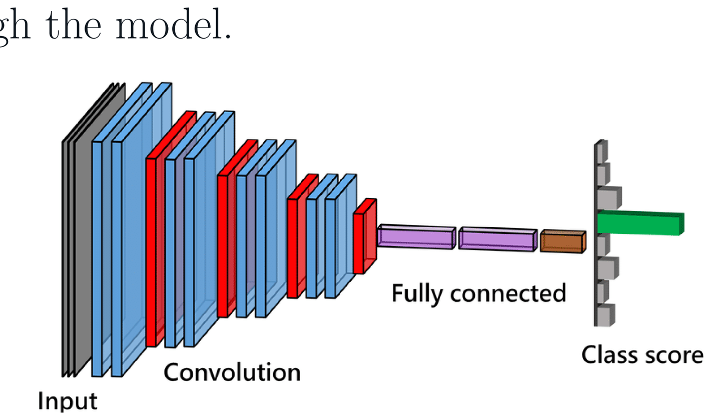

Fig. 3: Convolution Neural Network (CNN) [4]

Training

- During training, a model learns by going through each element of training data, evaluating its own output's accuracy against ground truth with a loss function, and then back-propagating changes to its algorithm based on calculated loss.
- Loss Function:

$$L = -\sum_{i=1}^{C} t_i log\left(\frac{e^{s_i}}{\sum_j^C e^{s_j}}\right)$$

C classes total; t_i, s_i are target and class score of class C_i

- Gradient-Based Optimization:

$$W = W - \lambda \nabla L$$

W are weighs; λ is the learning rate.

Image Classification

Image Classification: the task of assigning an input image one label from a fixed set of categories.

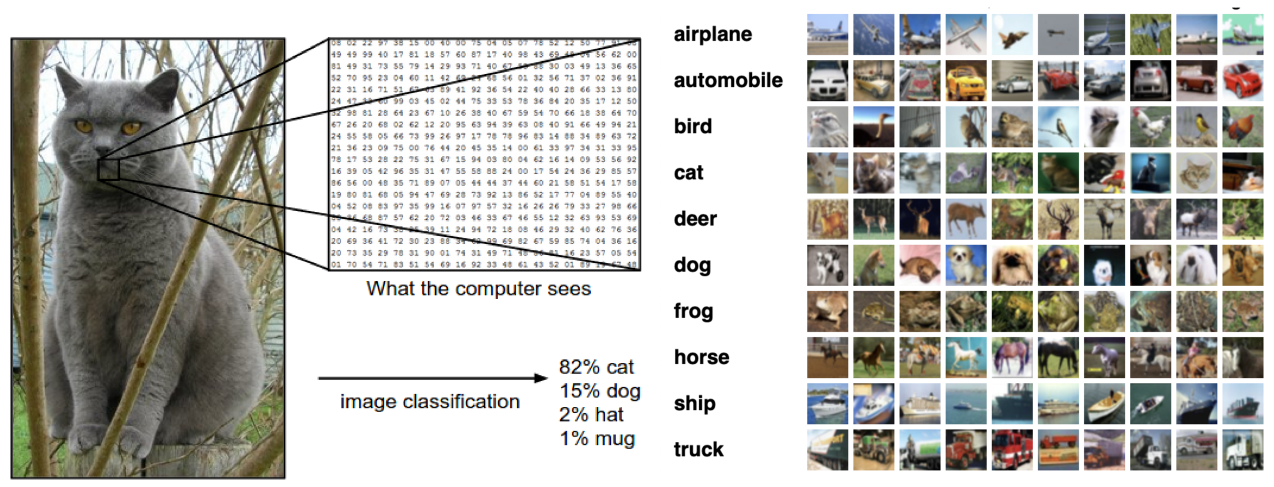

Fig. 4: **Left:**The task in Image Classification[5], **Right:** CIFAR-10 dataset[2]

Network Architectures

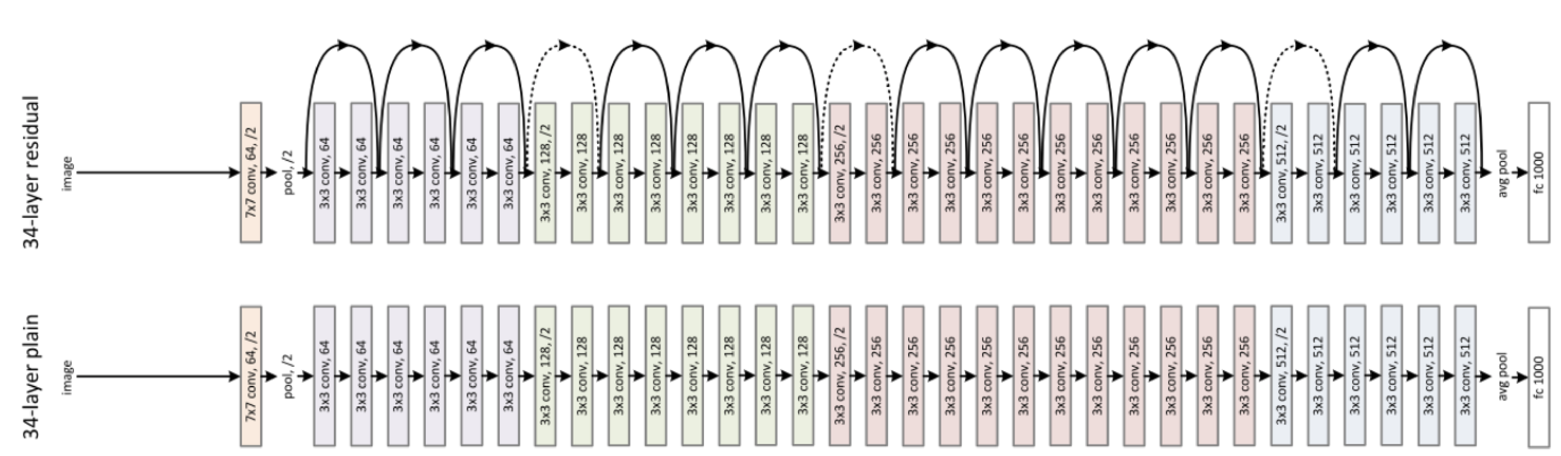

Fig. 5: Plain and Residual Network[1]

Residual Network is based on Plain Network with additional shortcut connections. The building block of the Residual Network is defined as:

$$y = \mathbf{F}(x, \{W_i\}) + x$$

Here x and y are the input and output vectors of the layers considered.

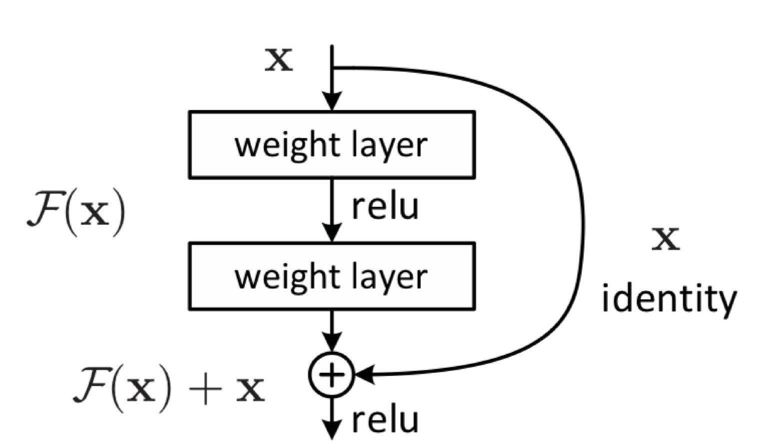

Fig. 6: Building Block of Residual Net[1]

Results

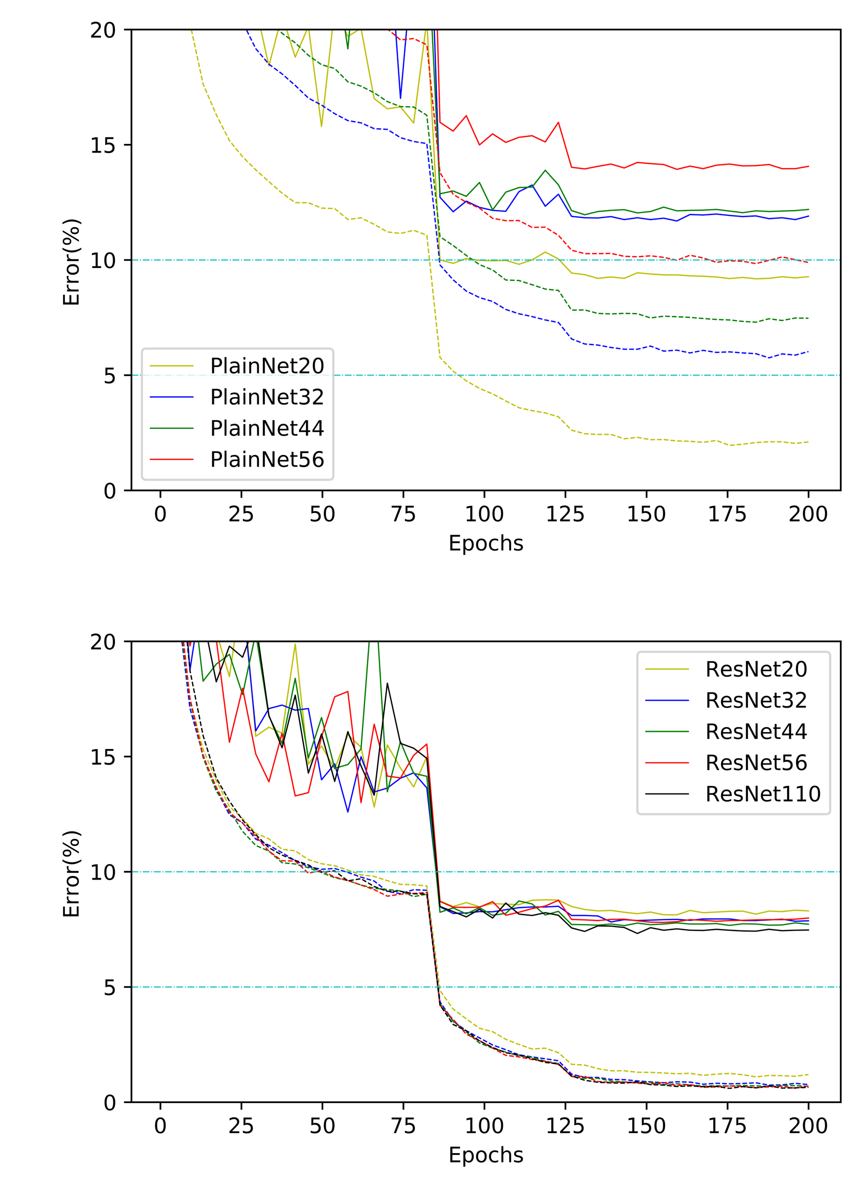

Fig. 7: Training on CIFAR-10. **DASHED** lines denote **trainning** errors, and **BOLD** lines denote **testing** errors. **Top**: Plain Networks. **Bottom**: Residual Networks

Comparison

Network	Training Accuracy(%)	Testing Accuracy(%)
20-layer ResNet	98.91	91.87
32-layer ResNet	99.28	92.18
44-layer ResNet	99.35	92.51
56-layer ResNet	99.39	92.26
110-layer ResNet	99.41	92.68
20-layer PlainNet	98.05	90.82
32-layer PlainNet	94.25	88.36
44-layer PlainNet	92.73	88.08
56-layer PlainNet	90.24	86.08

Acknowledgements

Jay Roberts and Kyle Mylonakis, our graduate advisers, for their help. UCSB Mathematics DRP for facilitating this opportunity.

References

[1] Kaiming He et al. *Deep Residual Learning for Image Recognition.* 2015. arXiv: `1512.03385 [cs.CV]`.

[2] Alex Krizhevsky. "Learning Multiple Layers of Features from Tiny Images". In: *University of Toronto* (May 2012).

[3] Nvidia. *Approaches to Object Detection.* URL: `https://www.slideshare.net/NVIDIATaiwan/nvidia-dli-approaches-to-object-detection`.

[4] Vladimir Puzyrev. "Deep learning electromagnetic inversion with convolutional neural networks". In: (Dec. 2018).

[5] Stanford. *Convolutional Neural Network for Visual Recognition.* URL: `http://cs231n.github.io`.