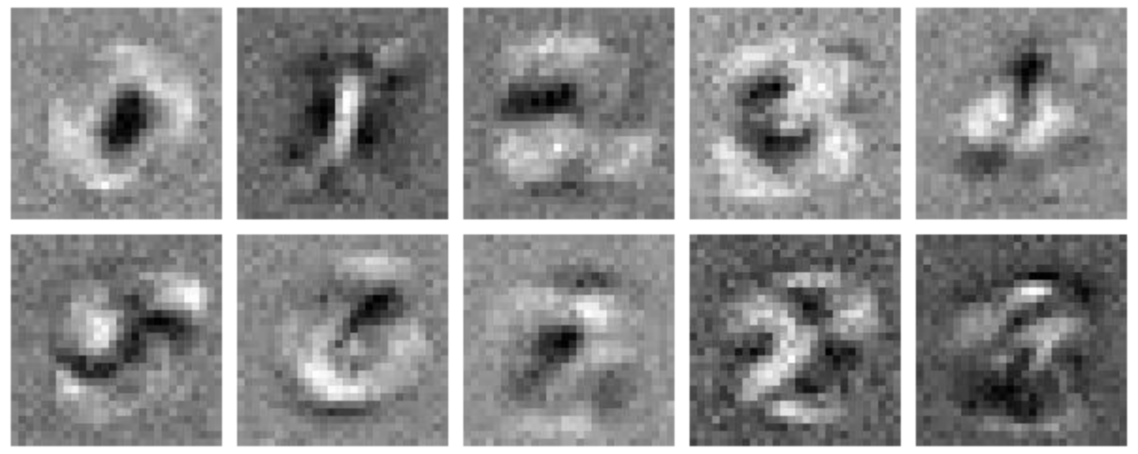
Visualizing Neural Networks

Mudasir Hanif Shaikh and Dr. Musabbir Majeed

Motivation

- Neural Networks have a lot of magic numbers called the hyperparamters, most of which have no set heuristics.
- Most deep learning frameworks hide the details of the network and there are a lot of things that happen under the hood.



MNIST digits seen by the deep neural network (from m4la.github.io)

Aims and Objectives

- Understand the heuristics of various hyperparameters such as the learning rate, batch size, regularization rate and how they affect the performance of neural networks.
- Extend these ideas to deep neural networks and their application in Computer Vision problems.
- Understand the working of tensorflow and use it for our computation.
- Implement the class visualization technique to understand how and what the neural network learns.

Methodology

Backpropagation Learning

Given a cost function,

$$C = \frac{1}{2n}\sum_x \|y - a^L\|^2 + \frac{\lambda}{2n}\sum_w w^2$$

the network learns using the backpropagation algorithm:

The error in the last layer: $\delta^L = \nabla_a C \odot \sigma'(z^L)$

The error in the subsequent layers: $\delta^l = \left((w^{l+1})^T \delta^{l+1}\right) \odot \sigma'(z^l)$

The gradients: $\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j$ and $\frac{\partial C}{\partial b^l_j} = \delta^l_j$

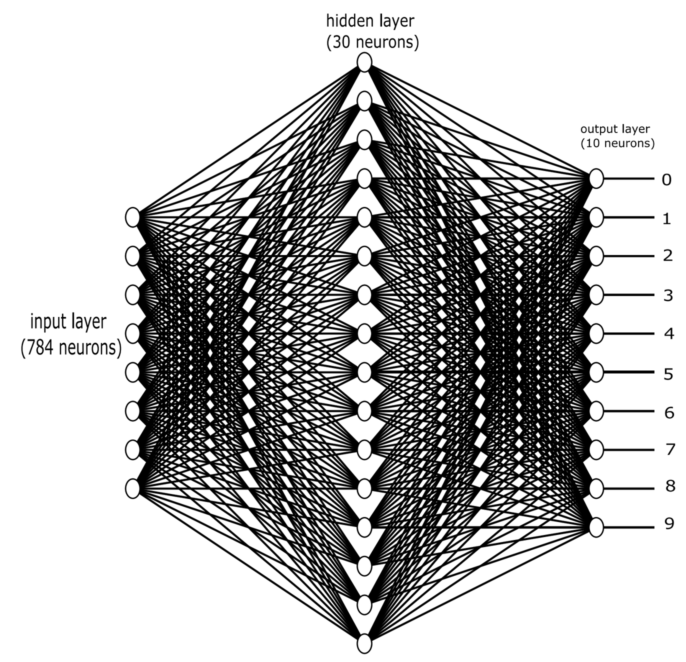
Class Visualization -- Deconvolution

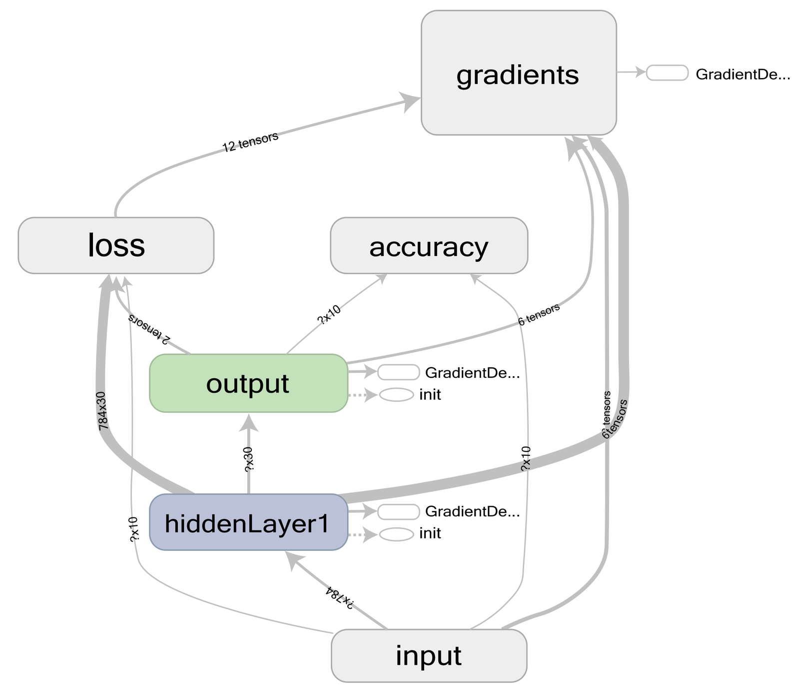
Applied deconvolution operation on pre-trained AlexNet.

$$I^* = argmax_I(S_c(I)) - \lambda||I||^2$$

Computational Graph – Tensorboard

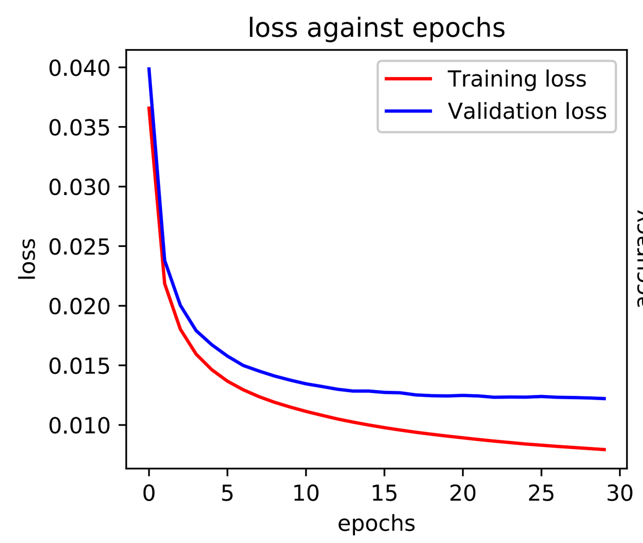
- Computational graph of a sample neural network produced using tensorboard.
- Input layer: 784 neurons; Hidden layer: 30 neurons; Output: 10 neurons

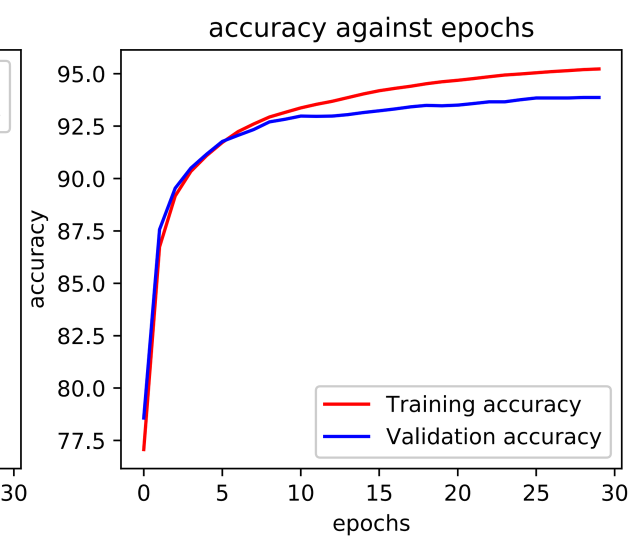




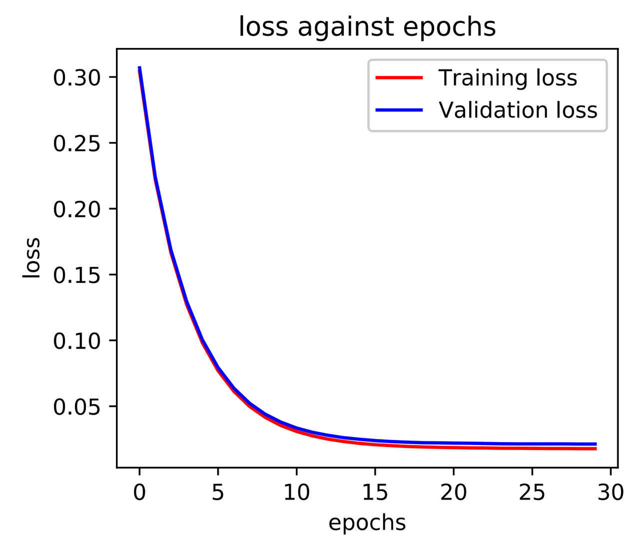
Results – Vanilla Network

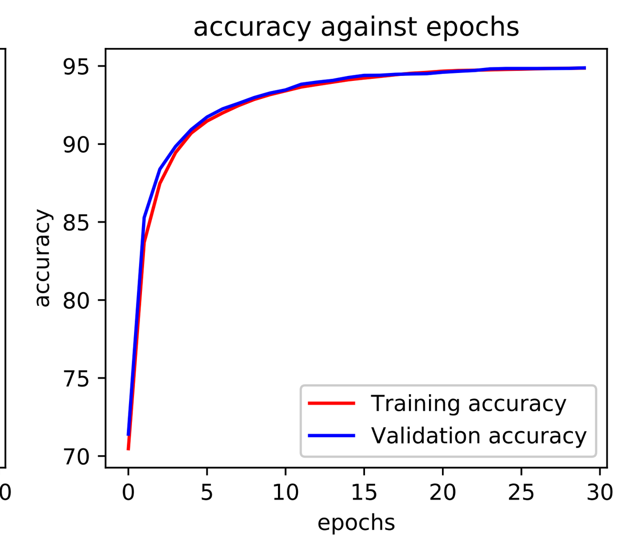
- Tested the network with following hyperparameters:
  learningRate(α): 2.5; batchSize: 50; iterations: 30



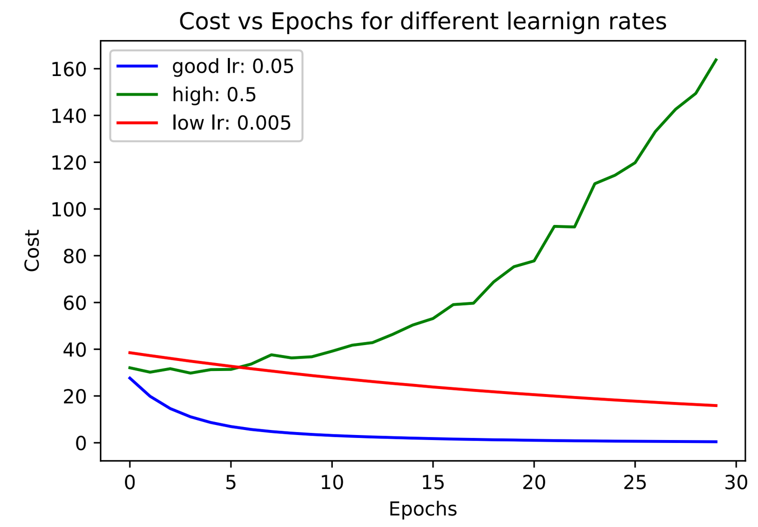


- In just 30 iterations, our simple model achieved a test accuracy of 94% on the MNIST digits dataset.
- Adding an L2-regularizor with λ = 0.001 reduced the gap between train and validation loss and raised the test accuracy by about 1% to 95%.



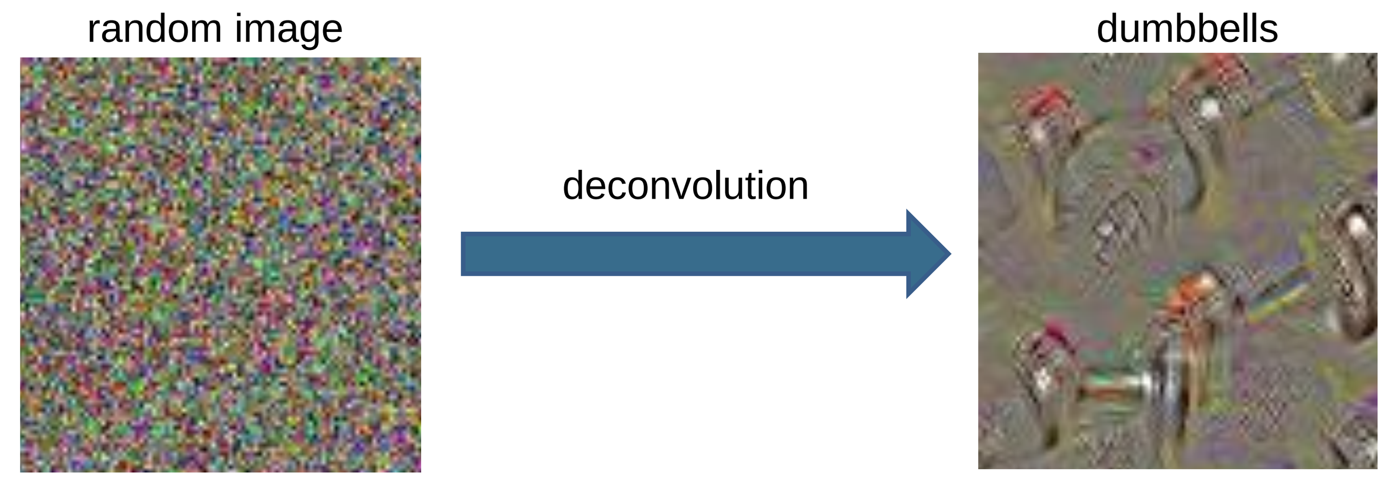


- The choice of hyperparameters (such as the learning rate(α)) is crucial to the performance of a network.
- A high learning rate may miss the minimum.
- A low learning rate takes a lot of time to converge.
- A good learning rate, however, converges fast.



Results – Class Visualization

- We start with a random image and perform gradient ascent with:
  learningRate:25; iterations: 5000; regularizationRate: 0.0001



Future Work

- Explore other visualization techniques to better understand what individual neurons learn.
- Build a tool that automates the visualization of the kernels, filters, and different layers of a network.
- Extend these ideas to sequential models such as RNNs and their application in language modeling.
- Build on these ideas to develop a debugging tool for neural networks which would help visualize the model to address machine learning issues such as exploding and vanishing gradients during training.

Habib University
shaping futures

Dhanani School of Science and Engineering