

Machine Learning

Analysis of Deep Learning Techniques



Mohit Jain - 201202164

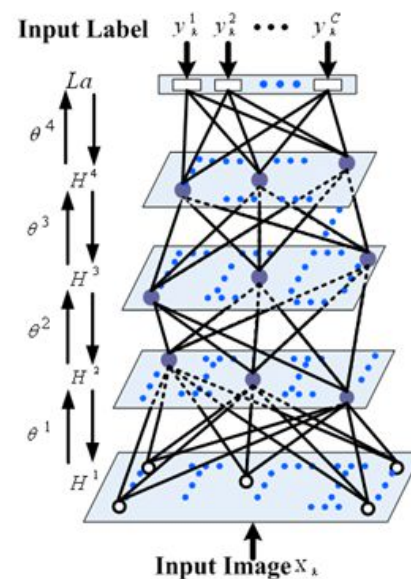
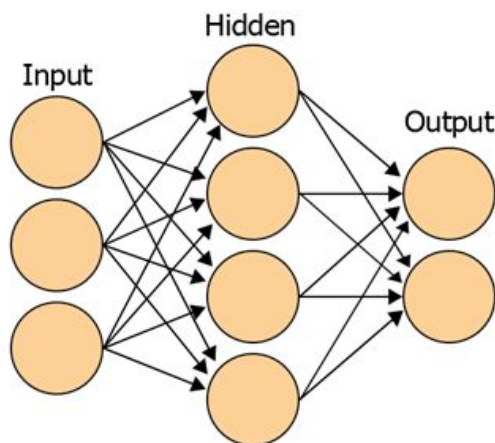
Fall 2015

Part-1

Success of Deep Learning

Deep Learning

Deep learning is the new big trend in Machine Learning promising general, fast and powerful machine learning algorithms, taking us a step closer to achieving artificially-intelligent machines. It's sudden fame in ML is due to some favourable results in applications where the target function is very complex and the datasets are extremely large.



An algorithm is “deep” if the input is passed through several non-linearities before being output, while "shallow" neural networks usually have one or two hidden layers. Most modern learning algorithms (including decision trees and SVMs and Naive Bayes) fall into the latter category.

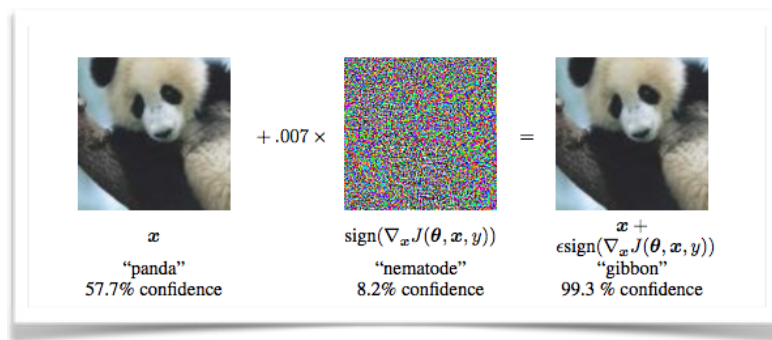
Convolutional Neural Networks (CNN) are biologically-inspired variants of the Multi-Layer Perceptron model. CNNs exploit spatially-local correlation by enforcing a local connectivity pattern between neurons of adjacent layers. In other words, the inputs of hidden units in layer m are from a subset of units in layer $m-1$, units that are sensitive to a spatially contiguous sub-space of the input space (receptive fields).

AlexNet

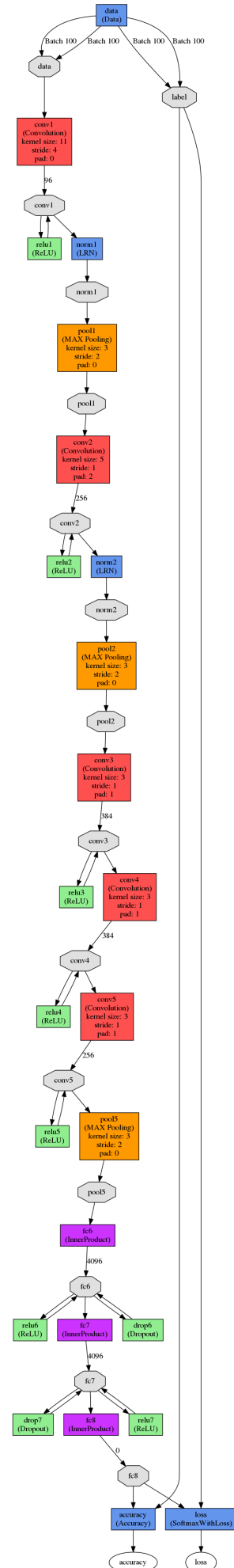
ImageNet Classification with Deep Convolutional Neural Networks - Krizhevsky et al.

AlexNet (named after Krizhevsky) is one of the state-of-the-art deep neural networks that came to fame in the ImageNet LSVRC-2010 contest. The network was used to classify the 1.2 million high-resolution images of ImageNet into 1000 different classes. On the test data, it achieved top-1 and top-5 error rates of 37.5% and 17.0% which was considerably better than the previous state-of-the-art. The proposed a modified version in the 2012 competition and achieved a winning top-5 error rate of 15.3%, half of the second position holder. The neural network has 60 million parameters and 650,000 neurons, and consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax for classification.

One of the major reasons for this networks outstanding performance was the incorporation of dropout (*setting to zero the output of each hidden neuron with a certain probability so that training is faster*) which showed the models dominance in terms of training efficiency.



However, recent work by Nguyen et al. and Szegedy et al. showed that AlexNet (or any other deep network for that matter) can be very easily fooled by adversarial training against such systems, creating very little perturbations in the inputs can cause a massive (near 99% error) change in the classification statistics.



Part-2

Convolutional Neural Networks

Cifar-10

Cifar-10 and Cifar-100 datasets are the smaller versions of 1.2 million high resolution ImageNet dataset. As the names go, they contain lower resolution (32x32) images for 10 classes and 20(coarse)/100(fine) classes respectively. I would be performing my experiments on the Cifar-10 and the fine version of Cifar-100. So lets get started!

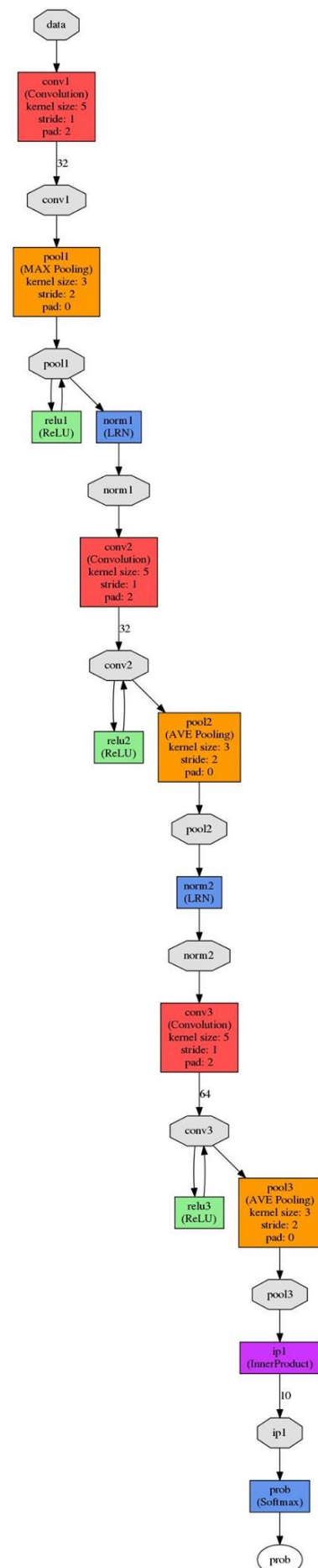
The network (lets call it CifarNet) has three iterations of the convolution-pooling-ReLU-normalise nodes with finally an inner-product layer to pass on the features to the softmax layer for classification.

Experiment 1 :

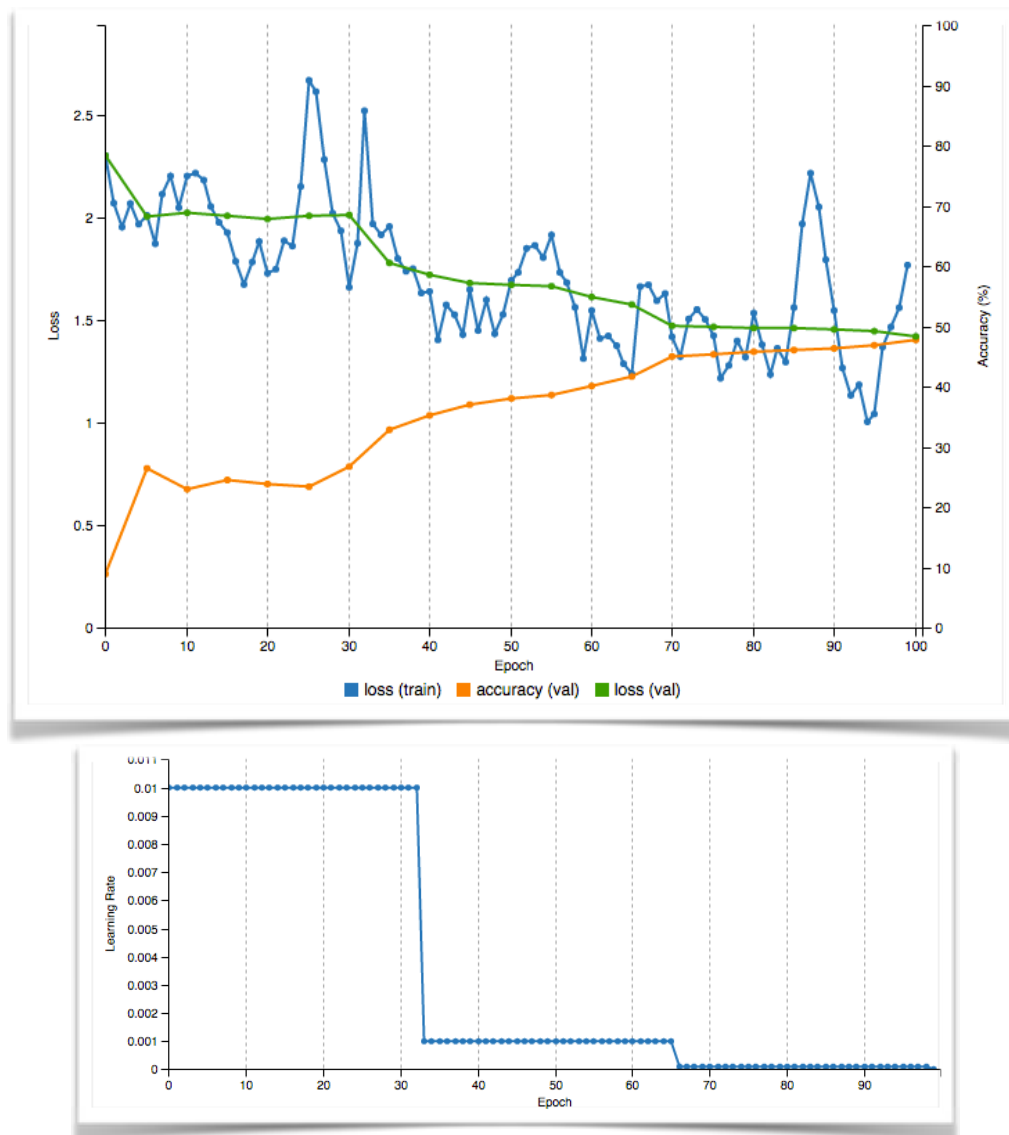
The first experiment conducted deals with using the raw image pixels as feature-inputs to CifarNet and the network is trained for 100 epochs. Model snapshots after every 10 epochs to see the progress of the training process were taken.

Experiment 2 :

The second experiment deals with using a machine learning trick used for optimization of the training of a network. The learnt parameters from the first experiment were used as input features to a SVM. The softmax layer of the CifarNet was done away with and the layer parameters from the last inner-product layer were given to the SVM as input.



The following are the results for **experiment 1**. In a different iteration from the one shown here, the CifarNet achieves 74.6% accuracy on the Cifar-10 dataset. The learning rate curves however are similar to the one's shown below.



The job took a total of 10 hours and 41 minutes to train for 100epochs.

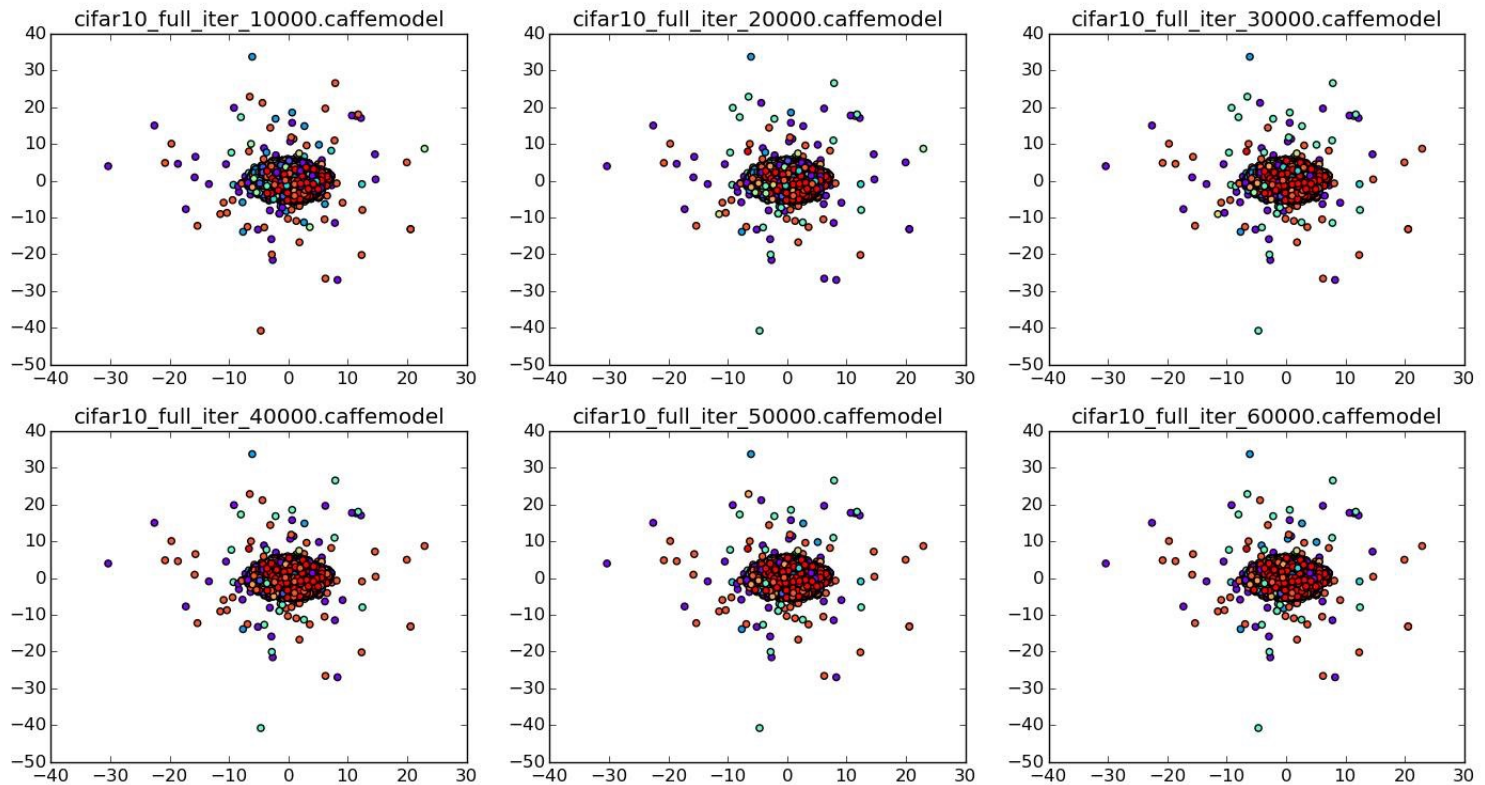
Job Status Done

- **Initialized** at Tue Nov 17, 07:26:34 PM (1 second)
- **Running** at Tue Nov 17, 07:26:36 PM (10 hours, 41 minutes)
- **Done** at Wed Nov 18, 06:08:19 AM (Total - 10 hours, 41 minutes)

Train Caffe Model Done ▾

Model training snapshots were taken every 10 epochs (10000 iterations through the dataset images) and tSNE visualisations for the same look as follows.

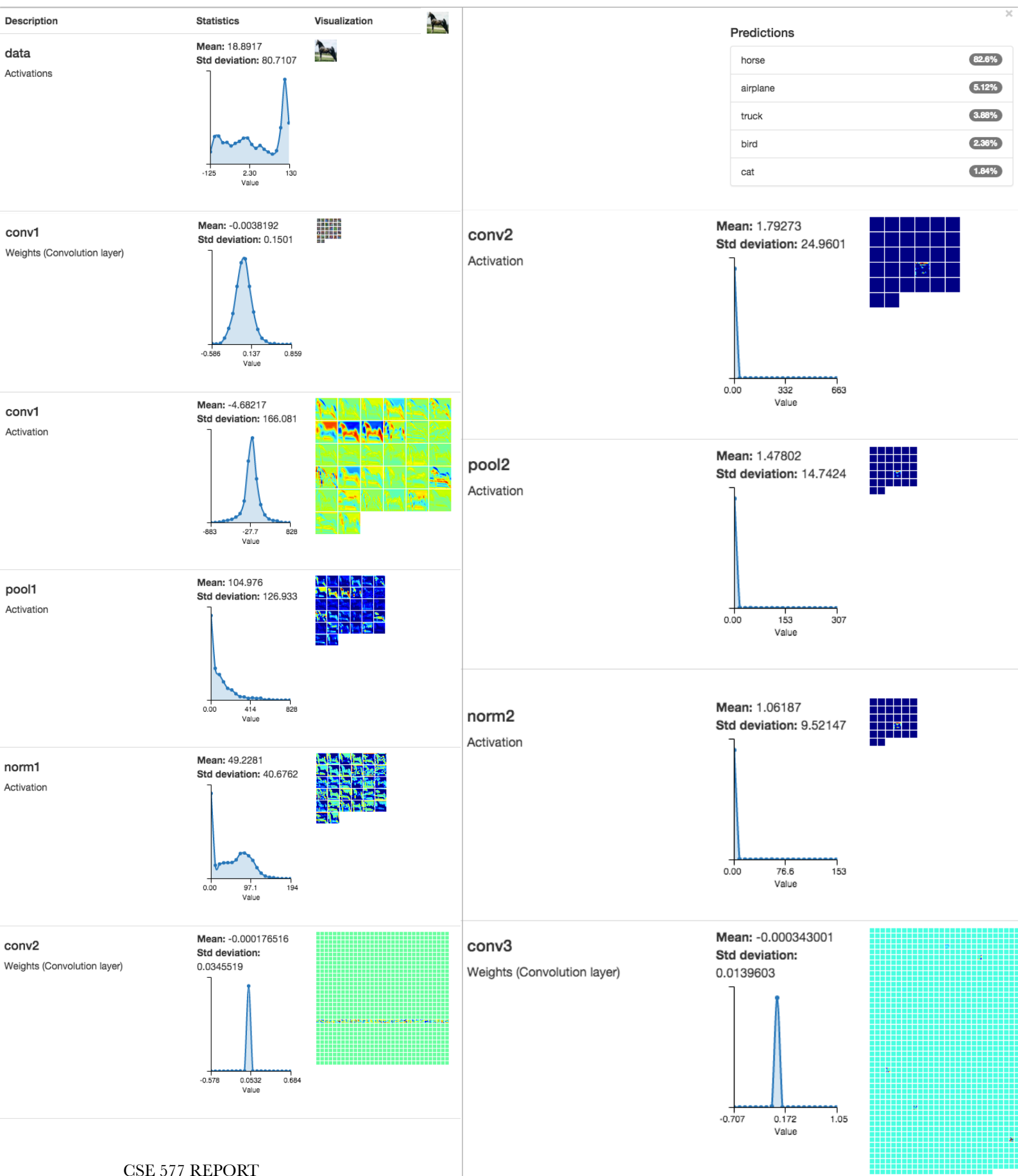
Decision Boundary updation as Training Progresses



The tSNE visualisations were computed using PCA to bring the 3-D images to lower dimensional manifolds and then computing the nearest neighbour techniques along with the tSNE algorithm to bring about the 2-D representation as shown above.

The visualisation show some progress in the classification bounds of the model but don't really shown distinctive boundaries due to the presence of outliers and not so perfect mapping to a lower dimensional manifold.

The following are the visualisations of the learnt layers for a sample ‘horse’ image on the CifarNet.



Experiment 2 :

The softmax layer of the CifarNet was done away with and the features from the second last layer of the network were given in to a SVM as input features to train upon.

We see that the SVM fails to learn the underlying boundaries between these feature points and performs miserably. The first intuition was that probably the datapoint were not normalised and hence the classification accuracy is as low as 7%. However, even after normalising the input features before training and testing, the overall precision turns out to be as low as nearly 10%.

SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0, degree=3, gamma=0.0, kernel='rbf', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False):

CifarNet-SVM	Precision	Recall	F1-Score	Support
0	0.1	0.2	0.15	1000
1	0.2	0.2	0.2	1000
2	0.05	0.1	0.07	1000
3	0.17	0.3	0.22	1000
4	0.03	0.1	0.07	1000
5	0.12	0.09	0.1	1000
6	0.14	0.1	0.12	1000
7	0.1	0.1	0.1	1000
8	0.23	0.21	0.22	1000
9	0.9	1.1	1.0	1000
Total/Average	0.1	0.2	0.15	10000

The precision-recall statistics clearly show that this techniques fails to be of any significance. The possible explanations for this can be the inherent dissimilarity between the shallow networks and deep networks way of computing features or the fact that Cifar-10/100 are very small and low resolution datasets and hence the features captured are not really well.

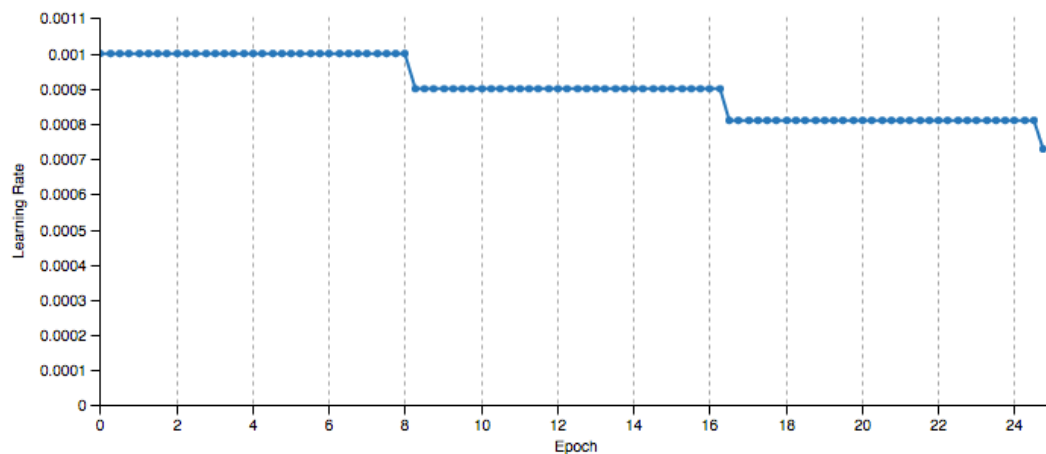
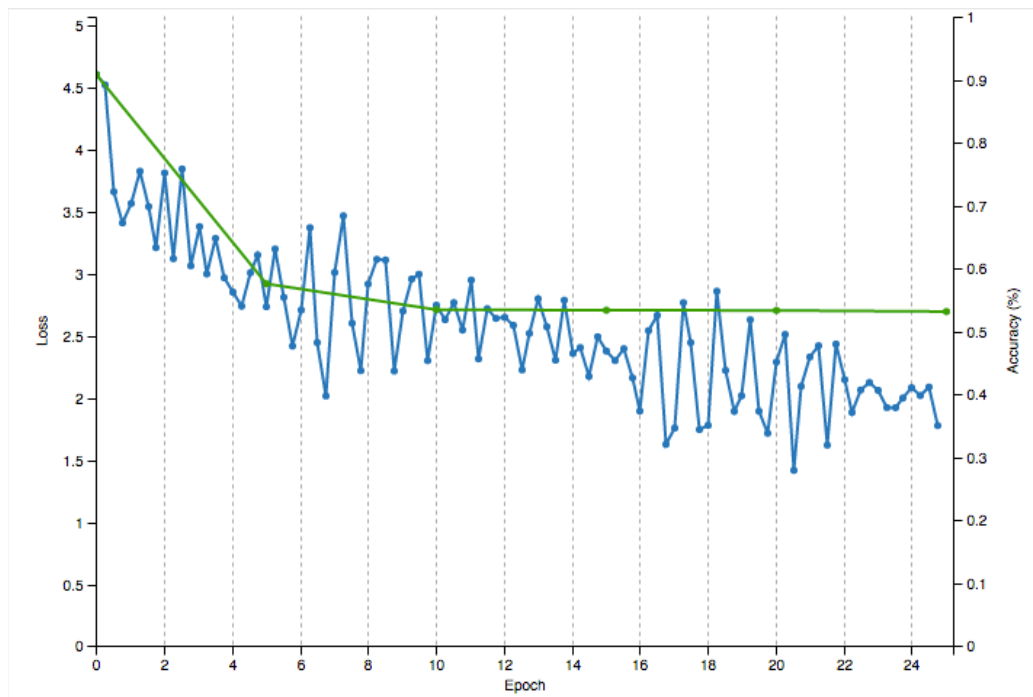
Part-3

Parameter Tuning

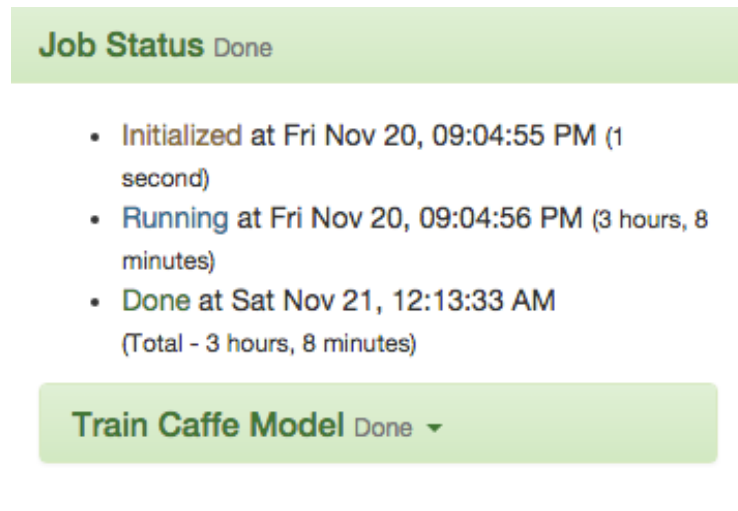
Cifar-100

This part majorly dealt with using trained features from the previous experiments to boost the training speed of the CifarNet modified to work on the Cifar-100(fine) dataset.

The softmax classifier and the last convolutional layer from the CifarNet of previous experiments were chopped off and a new fully connected layer and a softmax layer for 100 classes was added at the top of the model instead. The updated CifarNet was trained for 25 epochs.



The model took much less time to train (thanks to the optimization techniques used) and has a similar efficiency for classification as that of its predecessor.



The screenshot shows a user interface with two green buttons. The top button is labeled 'Job Status Done' and contains a bulleted list of training events. The bottom button is labeled 'Train Caffe Model Done' with a dropdown arrow.

Job Status Done

- **Initialized** at Fri Nov 20, 09:04:55 PM (1 second)
- **Running** at Fri Nov 20, 09:04:56 PM (3 hours, 8 minutes)
- **Done** at Sat Nov 21, 12:13:33 AM
(Total - 3 hours, 8 minutes)

Train Caffe Model Done ▼

Such techniques are widely used in machine learning experiments to speed up the process of training and evaluation and as seen from the experiment results, are highly useful.

Part-4

Advances in Convolutional Neural Network Architectures

ImageNet Classification

with Deep Convolutional Neural Networks

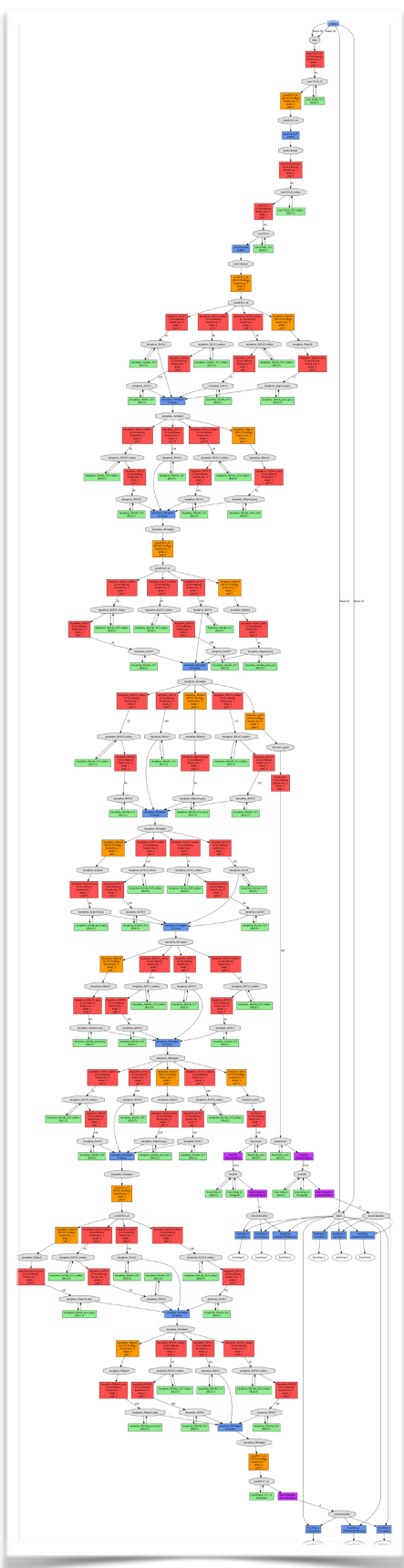
GoogleNet

GoogleNet is Scary. Yes, that's the word. The 22-layer deep giant convolutional neural network currently holds the championship title for the ILSVRC'14 (*ImageNet Large-Scale Visual Recognition Challenge*). While understanding the lower level details of this model is beyond the scope of this report, what one can really not miss is the manifold increase in the width of the model!

The idea of GoogleNet was to utilise optimally the computational resources at hand by incorporating multi-scale processing. It primarily focuses on classification and detection tasks and is the current state-of-the-art network for image classification and detection tasks. It achieves such high performance by using 2 iterations of the basic conv-pool-norm-ip nodes and then by introducing the “inception modules” which have dimensionality reducing nodes attached parallel to each other (increasing the width of the model) and a max-pooling layer to increase the number of outputs from stage-to-stage.

However, even GoogleNet suffers from the same adversarial fooling limitations shown by its predecessor Deep Networks mainly because of the inherent “smoothness assumption” that is taken in Deep Networks, which actually does not exist. These networks have “blind-spots” in the feature space which can be adversarially attacked.

One possible counter-measure is to escalate the feature-space to a non-linear distance metric where the adversary has no control on perturbations.



Thank You.

Submissions :

The experiments were performed using BVLC's Caffe and Nvidia's DIGITS. All scripts used in the process can be found at my GitHub page.

Link : <https://github.com/NightFury13/DeepLearningClassificationCifar>

A voiceover walkthrough of this report can be found at,

Link : <https://goog.gl/0qXpbi>