Fundamentals of Deep Learning for Computer Vision (by Nvidia)

What is AI, Machine Learning and Deep Learning

AI is getting machines to think like human (reasoning, thinking and making decision).Its the 4th revolution in Knowledge acquisition.

2 Types of AI

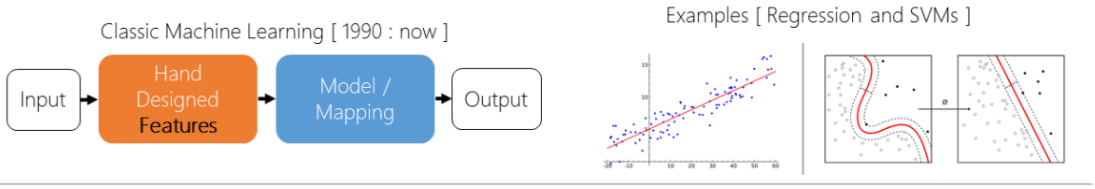
Narrow AI- like solving chess, or movie alphaGo

General AI- Training machines to think like humans

Machine Learning a subset of AI and a means to achieve AI

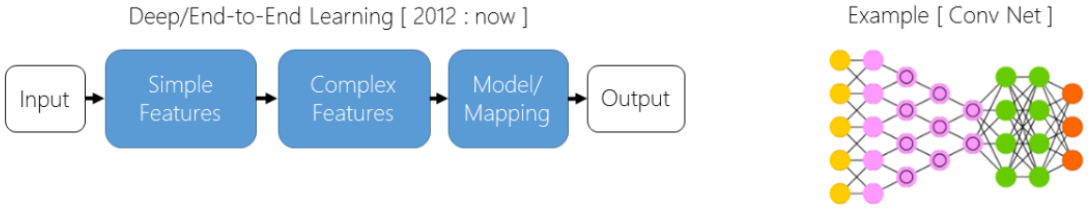
Deep Learning is the further subset and one form of Machine Learning which relies on the concept of Artificial Neural Networks

How trends to implement Machine learning have changed over the period specifically for image classification problems.



In past the major set back was to write down specific instruction or also called as an algorithm for the machine to understand. Consider an image classification problem to classify the correct number written in the picture how difficult it might get to write instructions for the same and this is just one for an image we might have N number of images and thus of instruction writing.

That's when Deep Learning comes into play, which allows machines to learn from data(examples). It identifies problem, find suitable examples that highlight both sides of the problem(the input and the output) and then a Neural Network bridges the gap.



<u>Real world applications of Deep learning</u>

1. Internet Services(Google Assistant, Siri)
2. Medicine(drug discovery, cancer cell detection)
3. Media & Entertainment(video captioning, content based search)
4. Security Defense(face recognition, cyber security)
5. Autonomous Machines(Lane tracking)

We will be using DIGITS environment for all our GPU tasks. DIGITS is not a framework it is a wrapper for NVCaffe, Torch and TensorFlow; which provides a graphical web interface to those frameworks rather than dealing with them directly on the command-line.

DIGITS can be used to rapidly train highly accurate deep neural network (DNNs) for image classification, segmentation, object detection tasks, and more. DIGITS simplify common deep learning tasks such as managing data, designing and training neural networks on multi-GPU systems, monitoring performance in real time with advanced visualizations, and selecting the best performing model from the results browser for deployment. DIGITS is completely interactive so that data scientists can focus on designing and training networks rather than programming and debugging

Deep Neural Network GPU task 1

The aim is to classify image to be Louie(dog name) or not Louie

1. We will choose a dataset which is designed for image classification Image of Beagles having 8 labeled images of Louie and other 8 of different dog.
2. Choose a Neural network we start with Alex Net the network which won ImageNet.

Observations with different architectures and changing one hyperparameters which is epochs recorded in an excel sheet.

Learnings:

- Learnt how to create model in DIGITS for dataset Image of beagles .
- Hyperparameter tuning is very simple by cloning the job and changing the value accordingly.
- Epochs effect on accuracy.
- You can classify an external image by simply pasting the image path or uploading an image

Deep Neural Network GPU task 2

In our Louie example, our model was effective on data from our dataset, but not on any new data, rendering it almost useless in the real world. So, we will be adding more data.

Two videos of a kid identifying or classifying violet, the kid couldn't identify the right features amongst all the inputted by his mom and thus the kid couldn't correct his model. The solution is to it more experience which in case of Deep learning is data.

We will be combining Deep Neural Networks, The GPU and Big Data to train a neural network that is effective for real world. Successfully training a neural network to perform well on new data requires enough data to demonstrate the diversity of the environment where your network should be effective.

In GPU task 1 we used a dataset with 16 images. So, we need more data and so we will be using Kaggle data set having 18750 labeled images of cats and dogs. Digits helps import the dataset and splitting into training and validation.

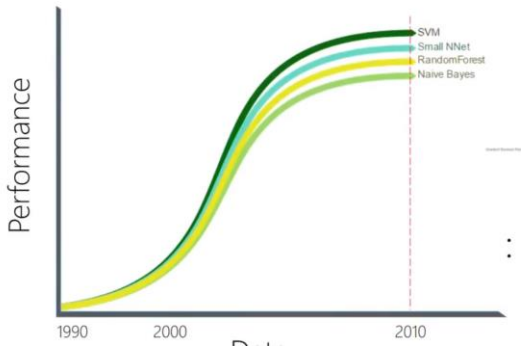
Observations with different architectures and changing two hyperparameters which is epochs and learning rate recorded in an excel sheet.

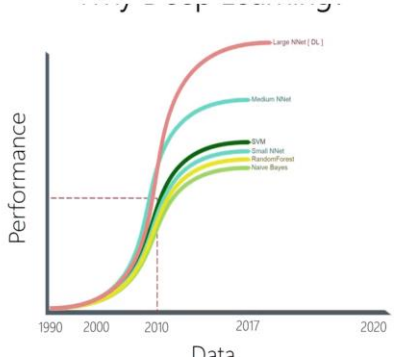
Learnings:

- How a larger dataset helps the model to learn better.
- When dataset with 18750 images the Alexnet model with 2 epochs gave a better performance than 16 images can see that in the comparison excel sheet. Also changing the learning rate affected its accuracy so probably we can increase the epochs to get the best accuracy.

- The pieces of deep learning that we have control over and that we don't
- The basic structure of a neural network

In the below picture we can how increase in data over the past 8 years have led to large increase in performance.





Deep Neural Network GPU task 3

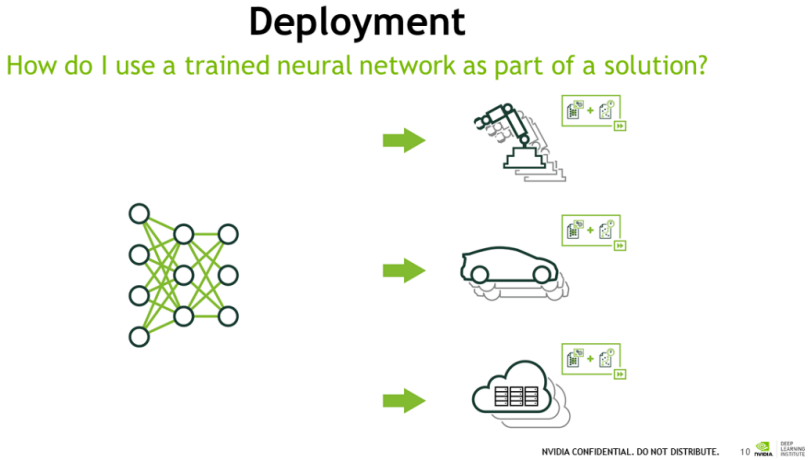
In the last two task we worked with image classification, where our deep neural network learned a mapping between an input of 256x256 color images to an output specifying the likelihood that each image belonged to a class, in our case dog or cat**.**

In this section we will learn to deploy trained networks into application to solve real world problems.

Problem Statement: We only allow dogs to go through the door and not cats

Solution: Deploy the trained dogs and cat model in the simulator.

Deployment is basically adding our model to applications like Robots, autonomous vehicles or on server to play a role in software.



To deploy our model 2 things are required which requires the code written for:

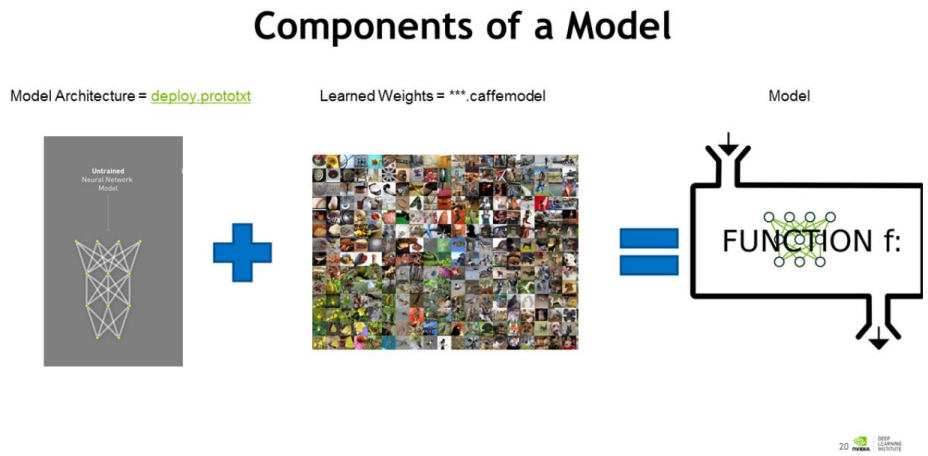
1. Model Input as expected.
2. Model output that is useful.

In our problem we will be taking the camera video input and output whether it's a dog and allowed to enter or a cat which is not allowed to enter the door.

1. Set the job directory for your model which will show you files in that directory
2. Our model consists of two files :
    - Architecture – the file for this is called as deploy.prototxt
    - Weights – the most recent snapshot file snapshot_iter_#.caffemodel
3. For basic deployment we need to install frameworks(caffe) and GPU for parallel processing to accelerate our operation speed on huge data.
4. Next is to create a classifier object(name it net) which takes in architecture file, weight file and information of data. This is done to access the model easily and make common action easy.
5. Preprocessing : creating an expected input a. Resize the image to 256 * 256 color images. b. DIGITS normalized the images by subtracting the mean image from each image to reduce the computation necessary to train.
6. Net object has a method predict which takes the normalized image and gives prediction. But the output is just a vector of length of two.
7. So, we generate a useful output through writing a python which checks the prediction argmax value and print the message as required accordingly and this process is called as postprocessing

Learning

- What a trained model contains



- Preprocessing(transform raw data to what network expects) and Postprocessing(prediction useful to user)

Deep Neural Network GPU task 4

In this section we will learn how to improve our accuracy. The model we trained was a classification for dog and cat and which matched with what human exactly a human would classify. So, the means to check if our classifier worked properly is the percentage accuracy or the percentage of images that were correctly classified. Main focus is to perform hyperparameter tuning and observe accuracy.

To save time we can save or model as pretrained model in DIGITS. Once we make a model as pretrained we can start our new model from where we left. Say for example I trained my model for Dogs Cats for 2 epochs and now I want my

model to run for 2 more epochs total 4 epochs and change its learning rate for same architecture, so I would use my pretrained model for first 2 epochs thus saving computation and training time.

There are four categories to improve performance :

1. Data : In early GPU task we saw that 18750 image datasets gave a good accuracy with less epochs so the more data the better the network.
2. Hyperparameters : include changing epochs, training time, architecture which is manual process. As we practice we will better understand what hyperparameters to tweak.
3. Training time : More epochs for some model but for some model it will cause overfitting.
4. Network Architecture : to solve a problem a specific architecture is required like for image classification Alexnet is considered so specifically to problems the architecture is chosen. Picking a right architecture requires a lot of practice.

Let's understand Hyperparameters epochs and learning rate which we have to tweak:

1. Epoch : A epoch is one complete presentation of data through network.
2. Learning rate : is the rate at which each weight changes during the training time. Learning rate decreases as the network gets closer to its ideal solution. Initially when data is given model doesn't know anything about the input but after few epochs it has understood rather learned the features of the data provided and thus the graph becomes stable or flat denoting that it has learned according to the input.

Deep Neural Network GPU task 5

In this section we will learn object detection, another computer vision problem which can be solved using Deep Learning. The most important thing while creating a model is to determine what our data is and what type of data we want.

The goal in this section is:

1. Whether an entity is present or not in the image
2. And if so where exactly it is

We implement 3 object detection using 3 approaches:

1. Sliding window
   We build the dog/not dog classifier
   This approach runs classifier on each 256X256 segment, we saw it identified the dog present in the picture to some extent not completely.
   Observing the execution time this approach is slow(constrained by image size) and needs human supervision
2. Modify Neural Network Architecture
   Layers in an architecture are mathematical operations on tensors(basically your input in array). Modifications to architecture impact the capability and performance of your model.
   In this approach we modified the Alexnet's architecture by replacing a convolution layer with some other convolutional layer .
   This gave a better result while detecting the dog as compared to the sliding window approach.
3. Detect Net
   We need labelled data. Labelling data is curating of input output mappings. In this case our input is image of any size and output is location of our object.
   The right network (or pretrained model is very important to build a solution for your problem.

Final Assessment

In this assessment we had to create a model for image classification to identify the face of whale or not.
Model Configurations
Number of epochs:- 5
Learning rate :- 0.0001
Optimizer:- Adam
Architecture:- Alexnet

Got model Accuracy : 98%

To deploy my model, I added architecture file deploy.prototxt and weights file to submission.py and condition to check if the face is whale or not
Then run the command to run the submission file to check if the image has whale and not whale.

Successfully deployed my model.