CS231n: CNNs for Visual Recognition **Spring 2017**

Lecture 9 & 10

Course Coordinator: Prof. Fei-Fei Li *Scribes: Akash Gupta*

CNN architectures -

1. AlexNet - 2012. Breakthrough result on ImageNet classification challenge (ILSVRC). Brought revolution of depth.
2. VGG & GoogleNet - 2014. Deeper Models. VGG-16, VGG-19. GoogleNet contains Inception modules that reduce the no. of parameters and apply convolutions in parallel. These models were released before Batch Norm and at that people used to train deeper models using some hacks. (Trained with less layers and then adding layers as the training progresses)
3. Resnet - Shortcut connections. Residual block tries to learn fewer at each block allowing for easier computation by the idea to only learn the perturbation and not the whole result itself (perturbation + identity).

   $$F(x) + x$$

   This will also drive the model to learn params in such a way that it will not use layers it does not need everytime and the corresponding residual block will simply give an identity output. Another advantage is it allows to solve the vanishing/exploding gradients since skip connections give a kinda straight highway for backprop to go back without going thru those many no. of computations.
4. DenseNet & FractalNet - Newer deeper models. More layers and skip connections.

Recurrent Neural Networks in Vision -

1. one to one $->$ Vanilla RNN
2. one to many $->$ Image captioning
3. many to one $->$ Sentiment classifcation
4. many to many $->$ Machine Translation
5. many to many $->$ Video classification

$->$ Sequential Processing of Non Sequence data - Generate images one piece at a time!

Truncated Backprop thru time - Run forward and backward thru chunks of the sequence instead of whole sqeuence. Carry hidden state forward in time but only backprop for some smaller no. of steps.

NOTE - Example of using char-RNN by Andrej Karpathy - min_char_rnn.py. Link

Image Captioning -

1. Take input an image by a CNN architecture. This will produce a summary vector from one of the FC layers at the end. Then feed it to an RNN by passing that vector as the initial hidden state (h_0) and a $< START >$ token to generate captions

Image Captioning with Attention - Tells the CNN model where to look to produce a specific word in the caption.

VQA: RNNs with Attention - Pass attention to the RNN specifying what part of the question to focus on. Then this allows the CNN architecture to focus on that specific part while summarizing the image.