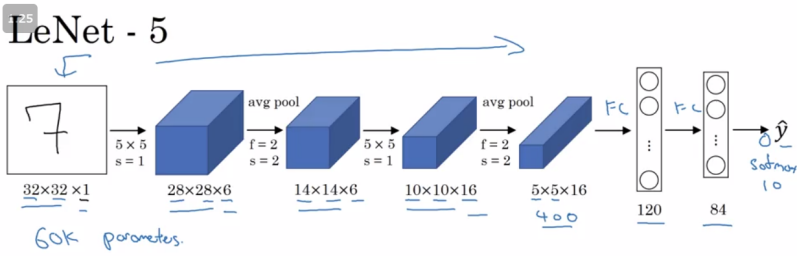
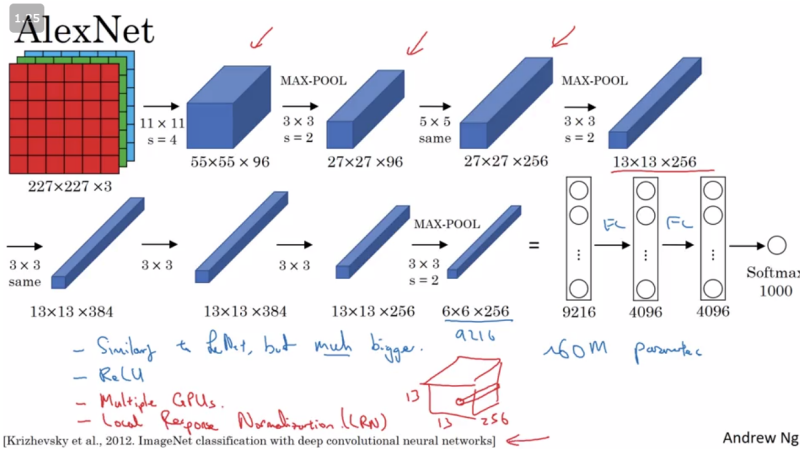
deep Learning_c4_week2

Created:	19/02/26 08:13	**Updated:**	19/03/01 20:19

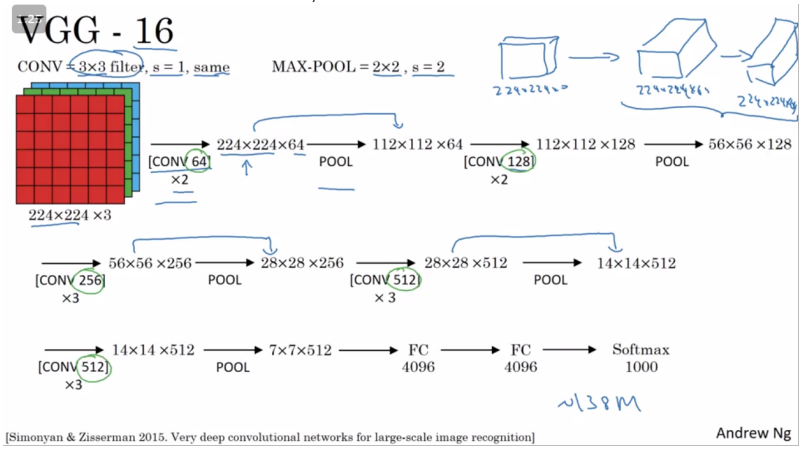
case studies

- classic network
  - three inspiring neural network
    - **LeNet-5**
      - publish in 1998
      - task: identify handwritten digits in a `32x32x1` gray image.
      - 

      - $n_H, n_W \downarrow \quad n_C \uparrow$
      - come up with **the popular architecture** : `Conv ==> Pool ==> Conv ==> Pool ==> FC ==> FC ==> softmax`
        - #when in 1998 softmax isn't widely used, the author used another method to make multi-task classification.
        - #at that time researchers always used average pooling.
        - basic idea: **N_h,N_W go down↓, and N_C go up↑**
    - **AlexNet**
      - publish in 2012
      - task: ImageNet challenge which classifies images into 1000 classes.
      - 

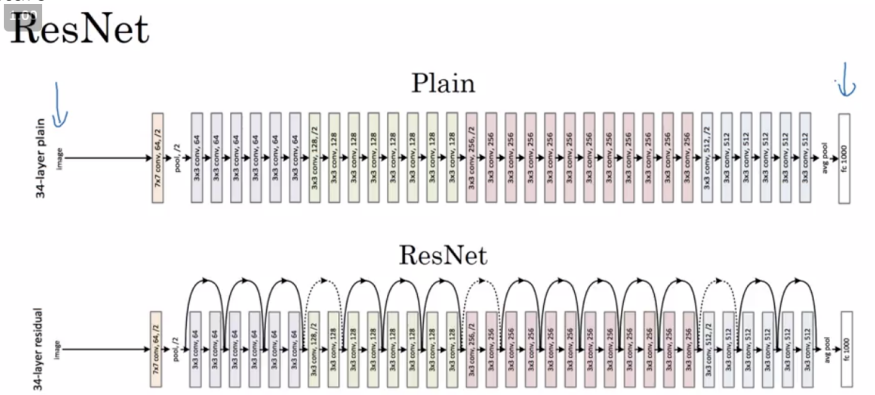
      - The original paper contains Multiple GPUs and Local Response normalization (LRN).
        - Multiple GPUs were used because the GPUs were not so fast back then.
        - Researchers proved that Local Response normalization doesn't help much
        - convinced deep learning is important in computer vision.

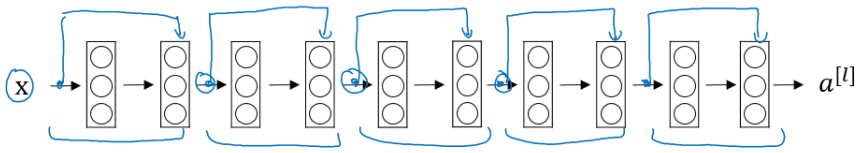
- **VGG-16**
  - publish in 2015
  - modern one with bigger architecture
  - **simple NN architecture**👉Focus on having only these blocks:
    - CONV = 3 X 3 filter, s = 1, same
    - MAX-POOL = 2 X 2 , s = 2



- **ResNet (Residual Network)**
  - design for **deeper** neural network.

    - In **theory**, as NN grow deeper the training error would go down.
    - However in **practice**, because of the vanishing and exploding gradients problems the performance of the network suffers as it goes deeper.
    - ResNet can solve such a problem 👇
  - architecture





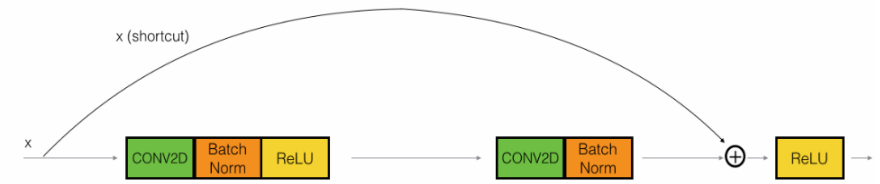


Figure 3 : **Identity block.** Skip connection "skips over" 2 layers.

- **skip-connection**
- `a[l+2] = g( z[l+2] + a[l] )`
- `= g( W[l+2] a[l+1] + b[l+2] + a[l] )`
- `if the dimension of a[l] and a[l+2] doesn't match`
  - `a[l+2] = g( z[l+2] + ws * a[l] ) # The added Ws make the dimensions equal`
  - ws also can be a zero padding.
  - or use 1x1 convolutions to match (which we would talk below)

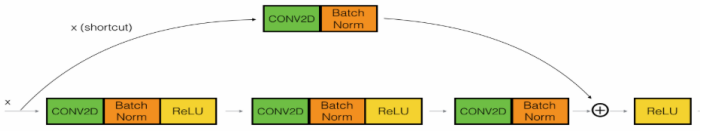
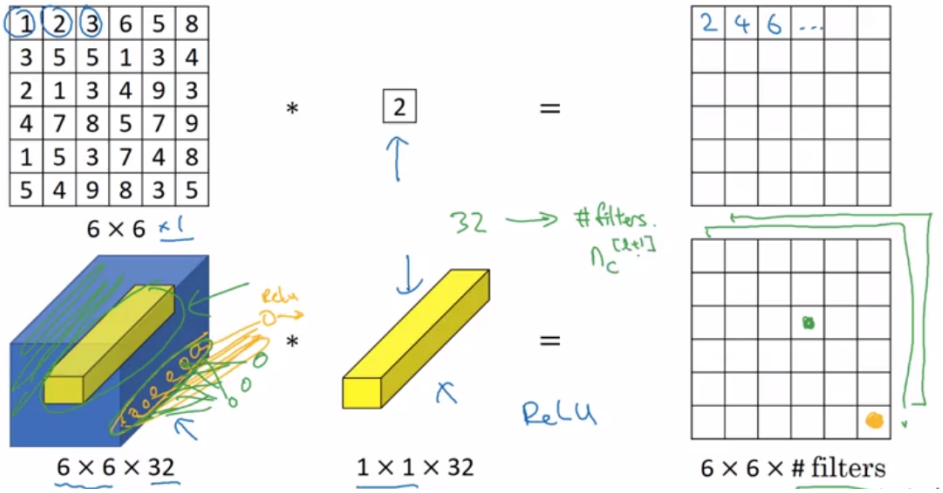


Figure 4 : **Convolutional block**

- why it work?
  - firstly, it would not hurt the plain NN performance 👇
    - if using L2 regularization, assume that it lead W and b equal to 0.
    - then `g( W[l+2] a[l+1] + b[l+2] + a[l] ) would equal to g(a[l])`
    - assume that we are using ReLU, then we have `g(a[l]) = a[l]`
  - secondly, it's likely to improve NN performance by learning more features from the residual layer.

Networks in networks and 1x1 Convolutions



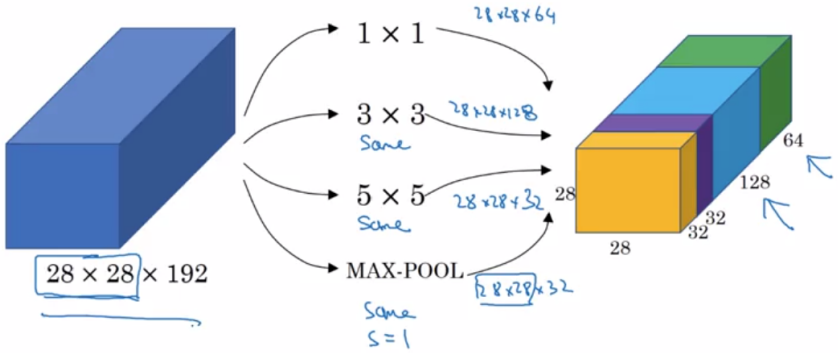
- quite useful when
  - specify a **exact** number of channels you want
  - **shrink** the number of channels to reduce computational cost

Inception network

- fun part: where does it's name come from?

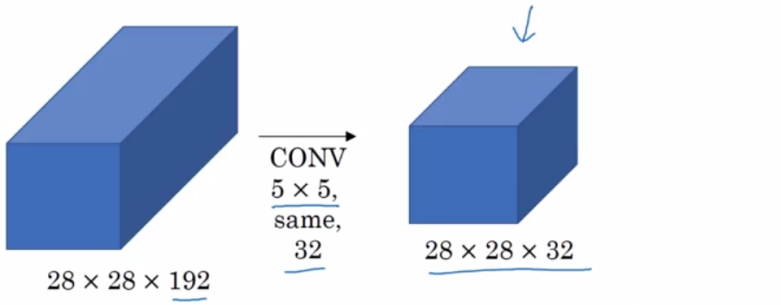
- http://knowyourmeme.com/memes/we-need-to-go-deeper Andrew Ng
- "don't hesitate for what to use, we implement it all LoL."
- basic blocks below👇

Motivation for inception network

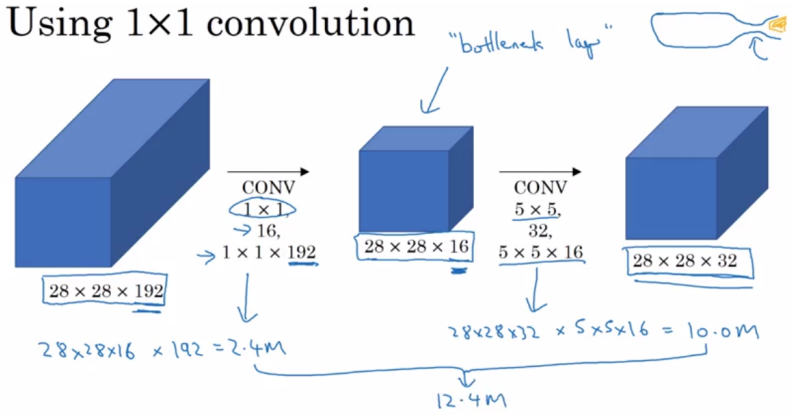


- trick: Using 1 X 1 convolution to reduce computational cost
  - A 1 x 1 Conv here is called **Bottleneck** BN
  - problem:

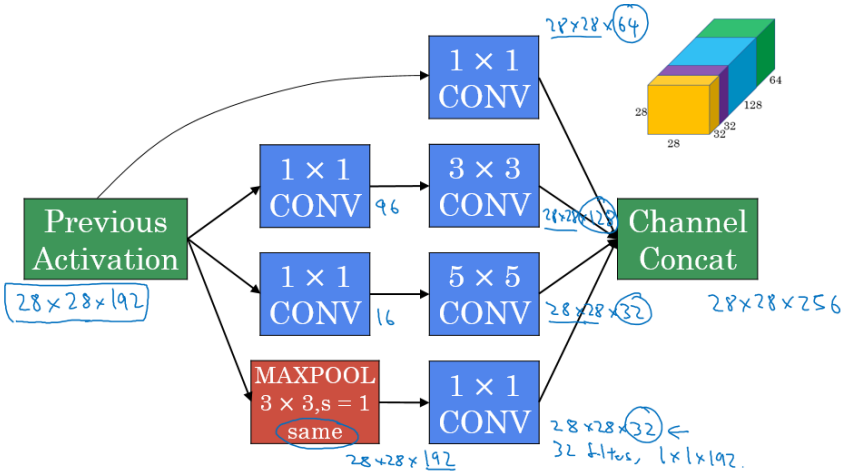
The problem of computational cost



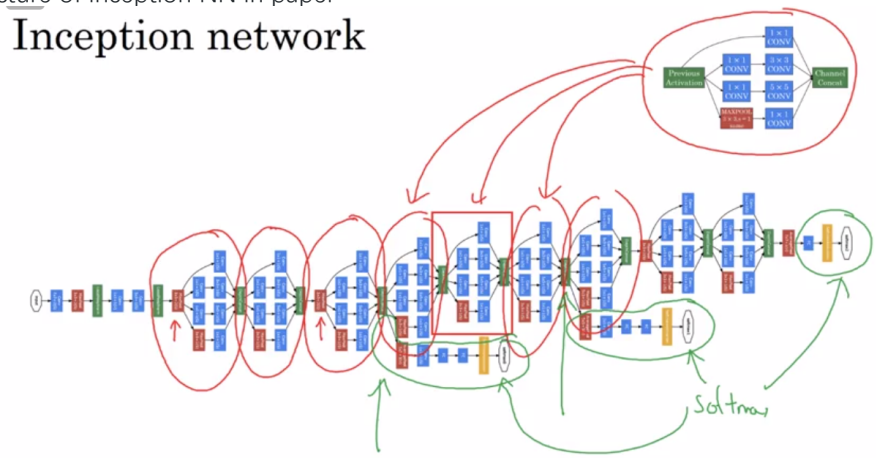
  - using bottleneck layer



- so now basic model could be:



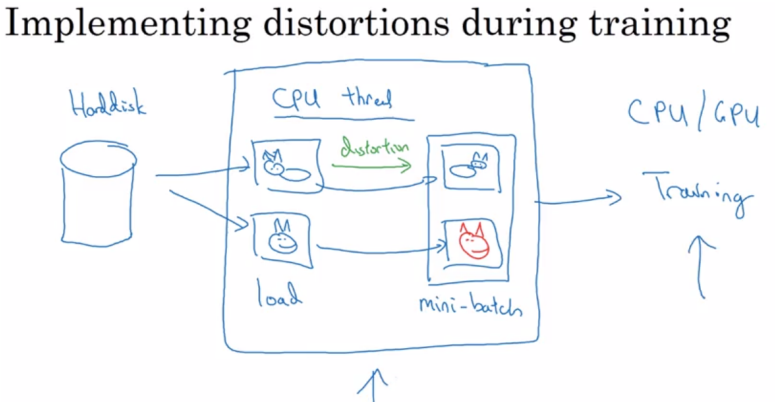
- architecture of inception NN in paper



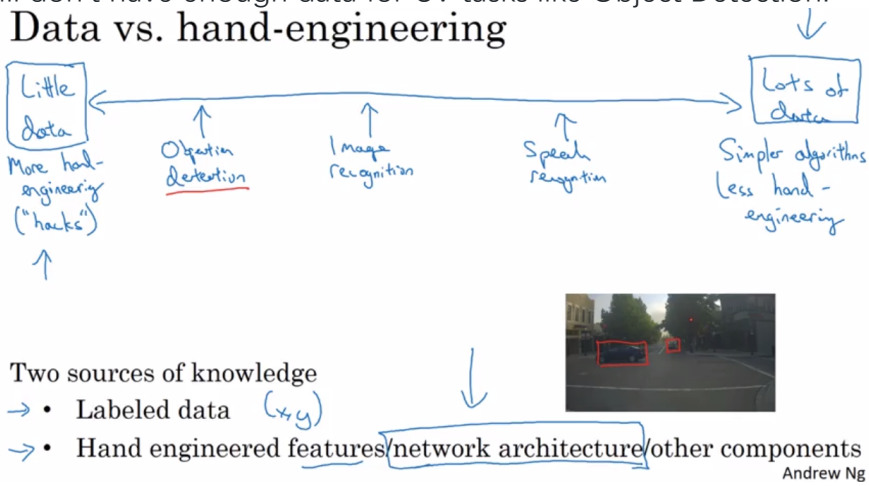
  - Max-Pool block (unit in red) is used before the inception module to reduce the dimensions of the inputs.
  - 3 Sofmax branches :
    - make prediction from hidden layers
    - helps to ensure that the intermediate features are good enough to the network to learn
    - it turns out that softmax0 and sofmax1 gives regularization effect.

practical advices

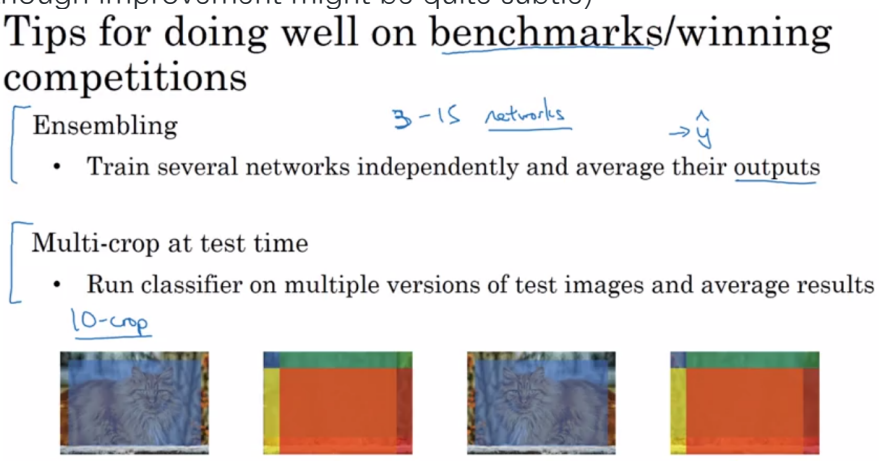
- using open-source implementation (like repo in Github)
- transfer learning
- data augmentation
    - methods:
        - rotation/mirroring/random cropping/shearing
        - color shifting_ ***PCA color augmentation***
        - Ex. parallel threads of DL



- state of CV
    - "we still don't have enough data for CV tasks like Object Detection."



    - tips (though improvement might be quite subtle)



    - use open source code

Use open source code

- Use architectures of networks published in the literature
- Use open source implementations if possible
- Use pretrained models and fine-tune on your dataset