Course-4 Week2

Deep convolutional models: case studies

Learn about the practical tricks and methods used in deep CNNs straight from the research papers.

Why look at case studies?

- We learned about Conv layer, pooling layer, and fully connected layers. It turns out that computer vision researchers spent the past few years on how to put these layers together.
- To get some intuitions you have to see the examples that has been made.
- Some neural networks architecture that works well in some tasks can also work well in other tasks.
- Here are some classical CNN networks:
  - **LeNet-5**
  - **AlexNet**
  - **VGG**
- The best CNN architecture that won the last ImageNet competition is called **ResNet** and it has 152 layers!
- There are also an architecture called **Inception** that was made by Google that are very useful to learn and apply to your tasks.
- Reading and trying the mentioned models can boost you and give you a lot of ideas to solve your task.

Classic networks

- In this section we will talk about classic networks which are **LeNet-5**, **AlexNet**, and **VGG**.

- **LeNet-5**

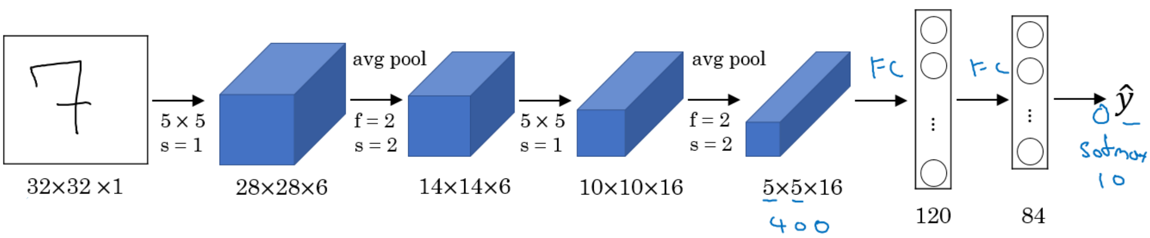
  - The goal for this model was to identify handwritten digits in a `32x32x1` gray image. Here are the drawing of it:

  - This model was published in 1998. The last layer wasn't using softmax back then.
  - It has 60k parameters.
  - The dimensions of the image decreases as the number of channels increases.

    - `Conv ==> Pool ==> Conv ==> Pool ==> FC ==> FC ==> softmax` this type of arrangement is quite common.
    - The activation function used in the paper was Sigmoid and Tanh. Modern implementation uses RELU in most of the cases.

- **AlexNet**
    - Named after Alex Krizhevsky who was the first author of this paper. The other authors includes Geoffrey Hinton.
    - The goal for the model was the ImageNet challenge which classifies images into 1000 classes. Here are the drawing of the model.
    - Similar to LeNet-5 but bigger.
    - Has 60 Million parameter compared to 60k parameter of LeNet-5.
    - It used the RELU activation function.
    - The original paper contains Multiple GPUs and Local Response normalization (RN).
        - Multiple GPUs were used because the GPUs were not so fast back then.
        - Researchers proved that Local Response normalization doesn't help much so for now don't bother yourself for understanding or implementing it.
    - This paper convinced the computer vision researchers that deep learning is so important.

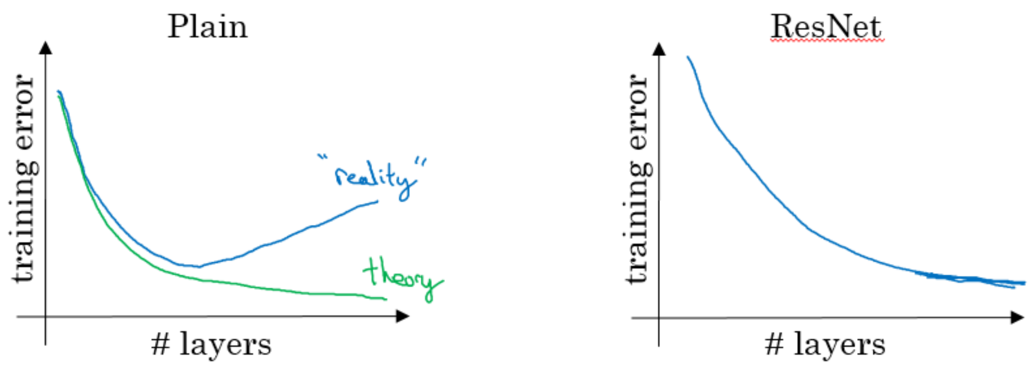
- **VGG-16**
    - A modification for AlexNet.
    - Instead of having a lot of hyperparameters lets have some simpler network.
    - Focus on having only these blocks:
    - It has a total memory of 96MB per image for only forward propagation!
    - Number of filters increases from 64 to 128 to 256 to 512. 512 was made twice.
    - Pooling was the only one who is responsible for shrinking the dimensions.



- There are another version called **VGG-19** which is a bigger version. But most people uses the VGG-16 instead of the VGG-19 because it does the same.
- VGG paper is attractive it tries to make some rules regarding using CNNs.

Residual Networks (ResNets)

- Very, very deep NNs are difficult to train because of vanishing and exploding gradients problems.
- In this section we will learn about skip connection which makes you take the activation from one layer and suddenly feed it to another layer even much deeper in NN which allows you to train large NNs even with layers greater than 100.
- **Residual block**
  - ResNets are built out of some Residual blocks.
  - They add a shortcut/skip connection before the second activation.
  - The authors of this block find that you can train a deeper NNs using stacking this block.
- **Residual Network**
  - Are a NN that consists of some Residual blocks?
  - These networks can go deeper without hurting the performance. In the normal NN - Plain networks - the theory tell us that if we go deeper we will get a better solution to our problem, but because of the vanishing and exploding gradients problems the performance of the network suffers as it goes deeper. Thanks to Residual Network we can go deeper as we want now.



  - On the left is the normal NN and on the right are the ResNet. As you can see the performance of ResNet increases as the network goes deeper.

Why ResNets work

- This show that identity function is easy for a residual block to learn. And that why it can train deeper NNs.

- Also that the two layers we added doesn't hurt the performance of big NN we made.

- Using a skip-connection helps the gradient to backpropagate and thus helps you to train deeper networks
- Residual blocks types:
  - Identity block:
  - The convolutional block:

Inception network motivation

- When you design a CNN you have to decide all the layers yourself. Will you pick a 3 x 3 Conv or 5 x 5 Conv or maybe a max pooling layer. You have so many choices.
- What **inception** tells us is, Why not use all of them at once?
- **Inception module**, naive version
- The problem of computational cost in Inception model:
- A 1 x 1 Conv here is called Bottleneck BN.
- It turns out that the 1 x 1 Conv won't hurt the performance.
- **Inception module**, dimensions reduction version:

Inception network (GoogleNet)

- The inception network consist of concatenated blocks of the Inception module.
- The name inception was taken from a *meme* image which was taken from **Inception movie**
- Here are the full model:
  -
- Some times a Max-Pool block is used before the inception module to reduce the dimensions of the inputs.
- There are a 3 Sofmax branches at different positions to push the network toward its goal. and helps to ensure that the intermediate features are good enough to the network to learn and it turns out that softmax0 and sofmax1 gives regularization effect.
- Since the development of the Inception module, the authors and the others have built another versions of this network. Like inception v2, v3, and v4. Also there is a network that has used the inception module and the ResNet together.