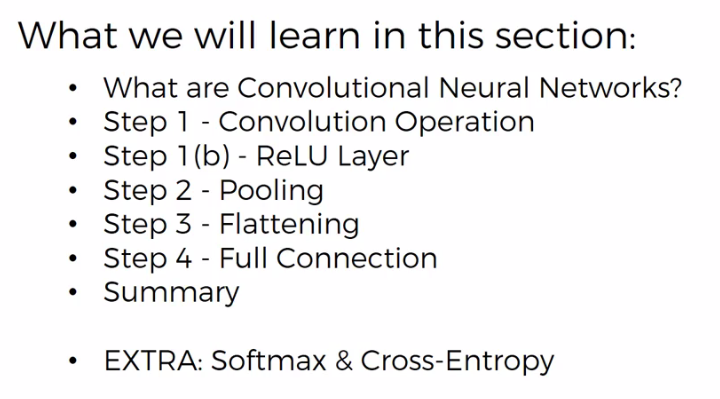
Chapter 9 : Part 1

**Deep Learning**

**CNN: Convolutional Neural Network**

Introduction to CNN

**9.1.0 Overview of what we will learn**



1. What Convolutional-Neural-Networks (CNN) actually are: We'll have a look at a few examples. We'll compare the human brain to Artificial Neural Networks in terms of Image Recognition.
2. Step 1 –Convolution Operations: This is a part of the steps to build a CNN. We'll learn about feature detectors, filters, feature maps, and the different parameters- what they mean and have a look at some visual examples.
3. Step 1 (b) – ReLU Layer: It is the Rectified Linear Unit (ReLU) and we'll talk about why *linearity is not good* and how we want *more nonlinearity* in our *network* for ***image recognition***.
4. Step 2 – Pooling: We'll understand how pooling works. We'll talk specifically about Max-pooling and also mean-pooling or sum-pooling and other approaches that you can take to the process of pooling. We'll see some example.
5. Step 3 – flattening: It's going to be a quick tutorial on how to proceed from your ***Pooled-layers*** to ***Flatten-layer***.
6. Step 4 - Full Connection: In this section we put everything together and everything into perspective. Here we will understand how everything works. How those final neurons understand how to classify Image.
7. Summary: Summarize everything we've talked about.
8. Softmax and Cross-Entropy: Not compulsory but these are terms that you will come across when dealing with CNN.

**9.1.1 Convolutional-Neural-Networks (CNN)**

|  |  |
| --- | --- |
| Look at this image. Do you see a person looking at you or do you see a person looking to the right. Here your brain is struggling to adjust, if you look to the right side of the image you'll see a person looking to the right. If you look at the left side of the image you'll see a person looking at you.   * This proves that when we see things is actually its *features*. Depending on the features our brain process. * So when you look on the *right* *side* of the *image* you see *certain features* of a person *looking to right* because they're closer to your center of focus and therefore your brain *classifies* as a *person looking to the right*. * When you look to the *left side of the image* you see more *features* of a *person looking at you* and therefore your brain classifies it as such. |  |

|  |  |
| --- | --- |
|  |  |
| * Most of this kind of illusion image we can see two in one and ***depending*** on which ***features*** our ***brain picks up*** it will ***switch*** ***between*** classifying each image as one or the other. The oldest one of these illusions recorded in the printed work is "duck or the rabbit". Here is also "Young girl or Old lady". * For this kind of image your brain is trying to understand what is it. what it is like it's trying to. This is a classic example of when there are certain features where your brain cannot decide. |  |

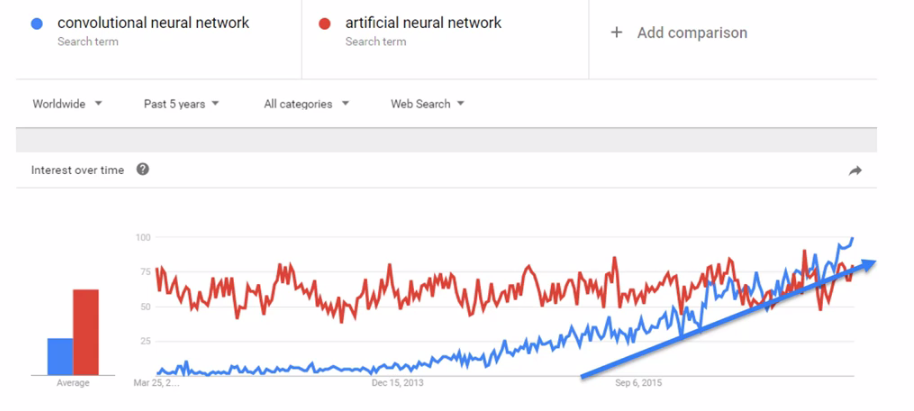
* All these examples illustrate to us *how the brain processes certain* features on an image or on whatever you see in real life and it classifies that as.
* You probably have been in situations when you ***look over your shoulder quickly*** and you see something it's like a Ball but it turns out to be a Cat because you don't have enough time to process those features or you don't have enough features to classify.
* The CNN works in very similar way. And computers can interpret them as we do.

|  |  |
| --- | --- |
| * Here's an experiment done on computers on CNN: here you see three images and we're going to go through them with left to right and see how you would classify then we see how computer classify. * So on the left you probably say Cheetah and computer said so (we're *going to learn how to read these images* because if you going to go deep into CNN we're going to start learning more and more and see a *lot of these kind of images*.) * In 2nd image the *neural* *network* was able to *distinguish* between *bullet-train/ passenger-car/subway-train/electric-locomotive*. |  |

* Actually there could be *many* *more* *options* and the *NN* learn to *distinguish* from those *categories* at the *same* *time*.
* For the *third* *image*, there are couple of options and it's not very clear what is it could be a *frying-pan/magnifying-glass/pair of scissors*
* You can see that the Probabilities are not as clear here so the neural network was a bit confused.
* Basically here you can see that *scissors* was its *first* *guess* but the correct option was number two and that's why it's *highlighted* in *red*.
* Later in this chapter we will learn, what these VOTES mean and how they are derived.

|  |  |  |
| --- | --- | --- |
| * How we read these image: So that's the actual correct label of the image "cheetah" in the ash color. That's the label of the images without any processing. |  |  |
| * And the computer vision (prediction) are here: the guesses the top four or five. They're given the probabilities so the computer said or the CNN said. * It said with a high probability "it's a cheetah" about like 95% or 99%. |  |

* CNN gained so much popularity over ANN. Because it is a very important field: that that is where all like *self-driving cars*, *recognize* *people* on the *road*, how to recognize *stop* *signs*, and things like that are build. How Facebook able to tag people in images also based on CNN.

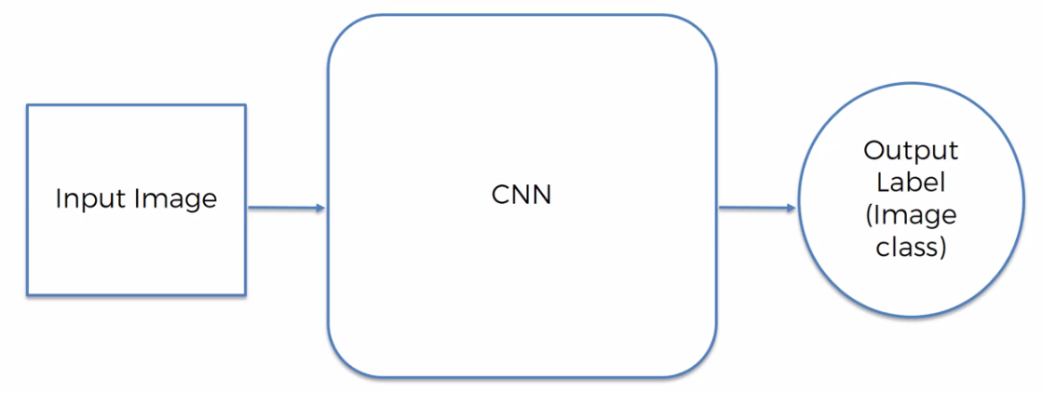


|  |  |
| --- | --- |
|  |  |

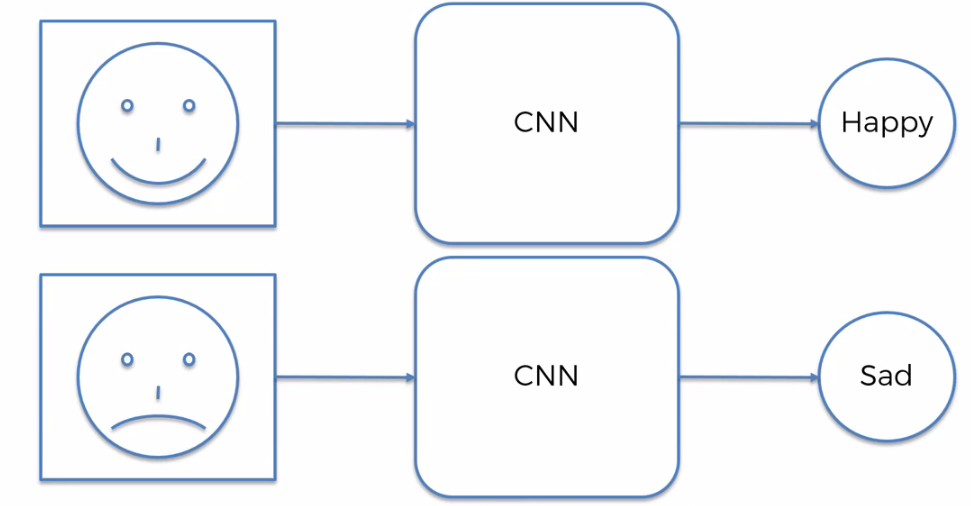
* If ***Jeffrey Hinton*** is the godfather of ***ANN*** and deep learning. Then ***Yann Lecun*** is the grandfather of ***CNN***. ***Lecun*** is a student of Jeffrey Hinton's.

**9.1.2 How CNN works**

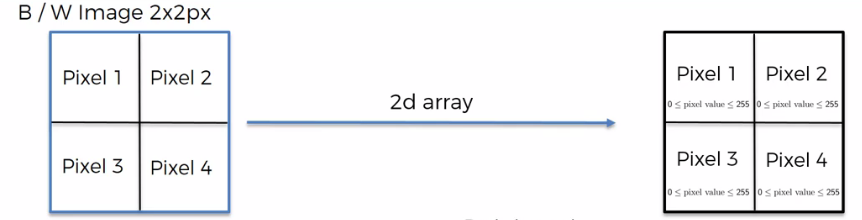
You have an input image it goes through the CNN and you then get a labeled-image as an output. So it classifies that image. As something like has a Cheetah or a Bullet Train or something else.



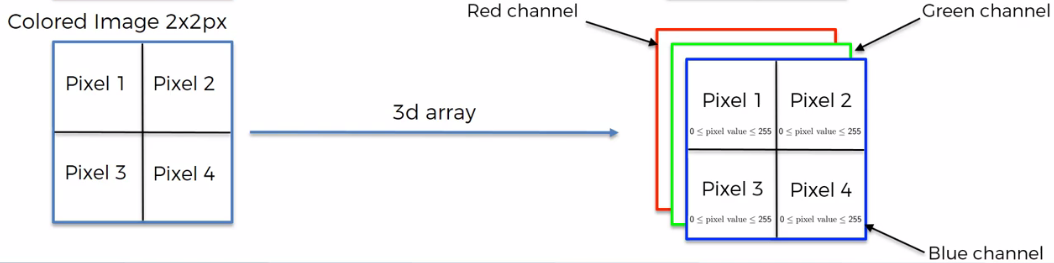
* A CNN can trained upon certain classified images and it can classify similar images from a given test-image. Let's say a *CNN* has been trained up to recognize *facial expressions*. You can give it a face of a *smiling person* and also a face of *sad person* and you train CNN from bunch of *pre-classified images*. Then after training the *CNN* can detect the *Happy/Sad* person from a *test* *image*.
* CNN gives you the *probability*, for example *85% chance* *of happy-Person* or *95% Sad*.
* And CNN can get confused sometime (as we get for some image/illusions).



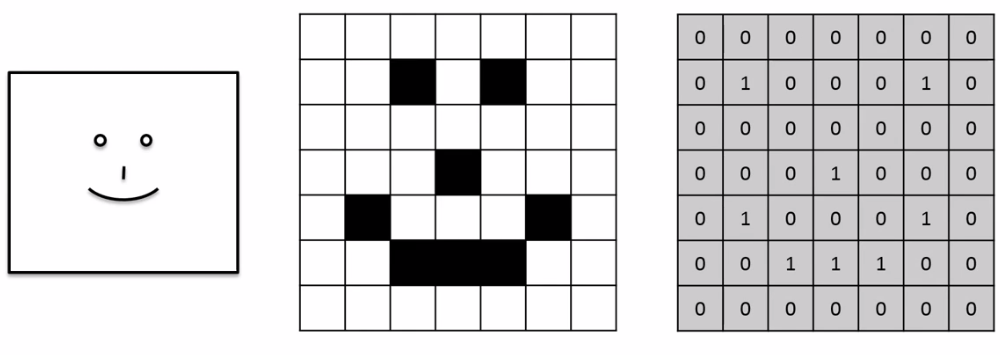
* How does CNN able to recognize these features: Let's say you have two images one is *black and white image* of ***2x2 pixels*** and Other one is a *colored image* of ***2x2 pixels***.
* Black & White - image: NN takes the black and white image is a two dimensional array with each of those pixels having a value between ***0*** and ***255*** (that's 8 bits of information .).
* The values from ***0*** to ***255*** and that's intensity of the color, ***0 = black pixel*** and ***255 = white pixel*** and between them you have the grayscale range of possible options for this pixel.
* And based on that information computers are able to work with the image as: Any image is actually has a digital representation/digital-form. And those are just basically ***1*** and ***0*** that form a number ***0*** to ***255*** for every ***single*** ***pixel***.
* It ***doesn't*** actually work with ***colors*** or anything, it works with the ***1***'s and ***0***'s.



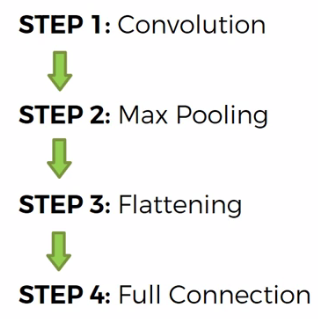
* Colored image: And in a color image it's actually a 3-dimensional array. You've got in ***RGB***, ***Blue-layer***, ***Green-layer*** and the ***Red-layer***. And *each* *one* of those *colors* has its *own* *intensity*.
* So basically a ***pixel*** has ***3 values*** assigned to it. Each one of them is between ***0*** and ***255***. Computers are going to work by combining those three values.
* Those are the ***red*** ***channel***, the ***green*** ***channel***, the ***blue*** ***channel***.



|  |  |
| --- | --- |
| * Example: Let's have an example of a smiling face. If we simplify things, instead of having values from ***0*** to ***255*** if we use only ***1***'s and ***0***'s, ***0 = white*** and ***1 = black***, then we can see that that image can be represented as below. * In this chapter we will more discuss about images like this, which is very simple having only ***1***'s and ***0***'s, but at the same time all those concepts can applied to the ***0*** to ***255*** range of values. |  |



* Steps to process images in CNN: And the steps are we're going to process these images are:



* Paper (Additional reading): This is the Yann LeCun's original paper that gave rise to CNN. It's called "Gradient Based Learning Applied To Document Recognition". If you want to go back to the very beginnings of how it all happened, where it all came from this is the paper to look into.

