PROJECT AI CONVOLUTIONAL NEURAL NETS THEORY

Feed-forward neural networks most closely match our description above: you have some input, a series of layers of artificial neurons, and ultimately some output. For example, we might turn last quarter's sales numbers into a projection for the areas to target in the next quarter.

Convolutional neural networks are a class of feed-forward neural networks specifically inspired by the visual cortex, and are particularly well-suited to recognizing visual input. These types of neural networks are powerful when the entire input is available at once, as with an image; think machine learning-generated visual artwork. They can also process medical images or visual road data. Limitation is it needs an image to work well

Recurrent neural networks, on the other hand, organize these layers into internal cycles, which allows these networks to process input where time is a factor. This is what allows recurrent neural networks to work in areas like speech recognition and natural language processing: they begin working as soon as input starts coming in rather than waiting for the speaker to finish their sentence.

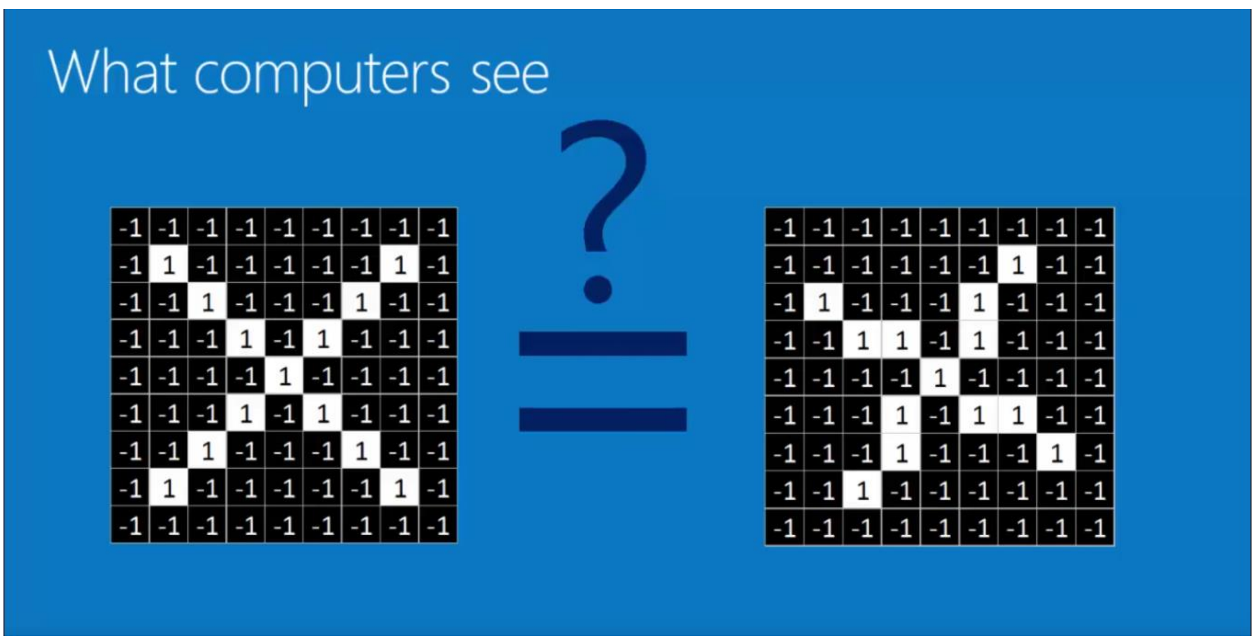
Challenges of CNN's in image classification

Translation (position of the handwritten item may vary)

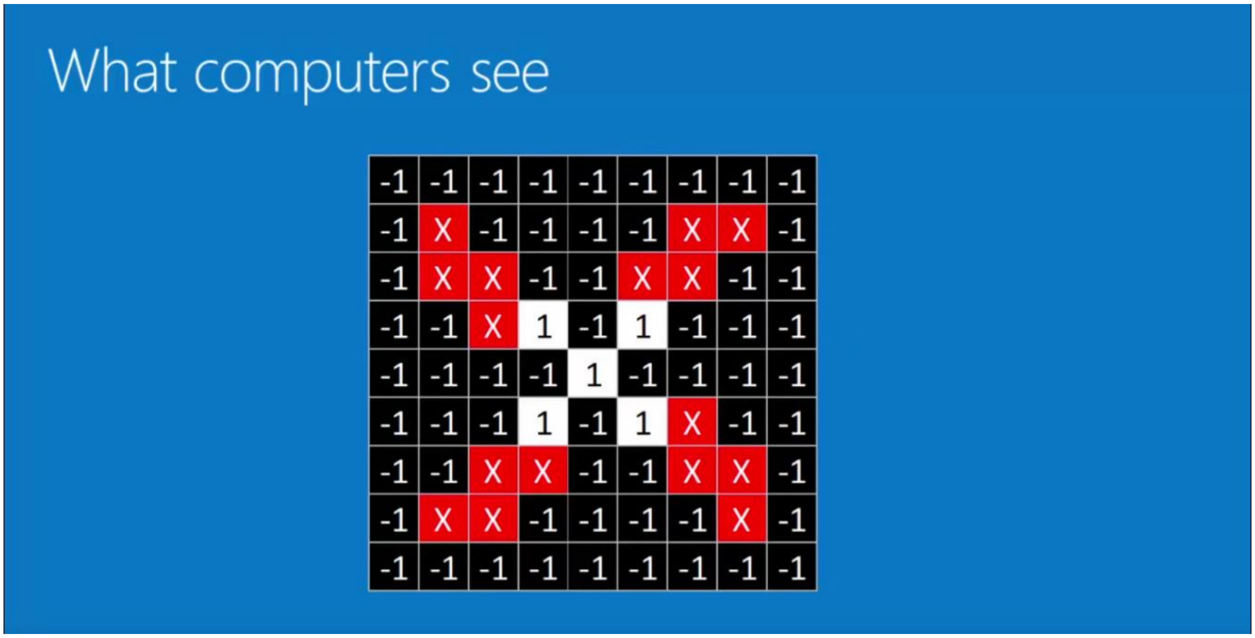
Scaling (larger or smaller)

Rotation (Might be tilted)

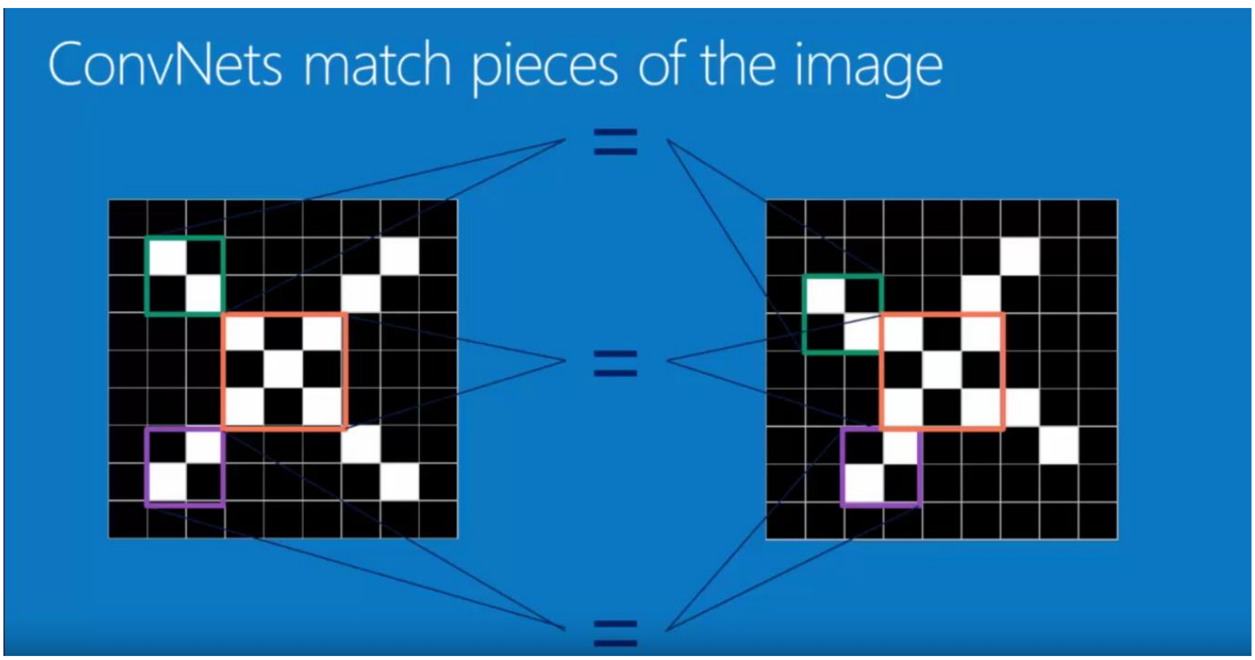
Weight (User may draw thick or thin lines)



TRY TO MATCH THESE 2 IMAGES



NO MATCH

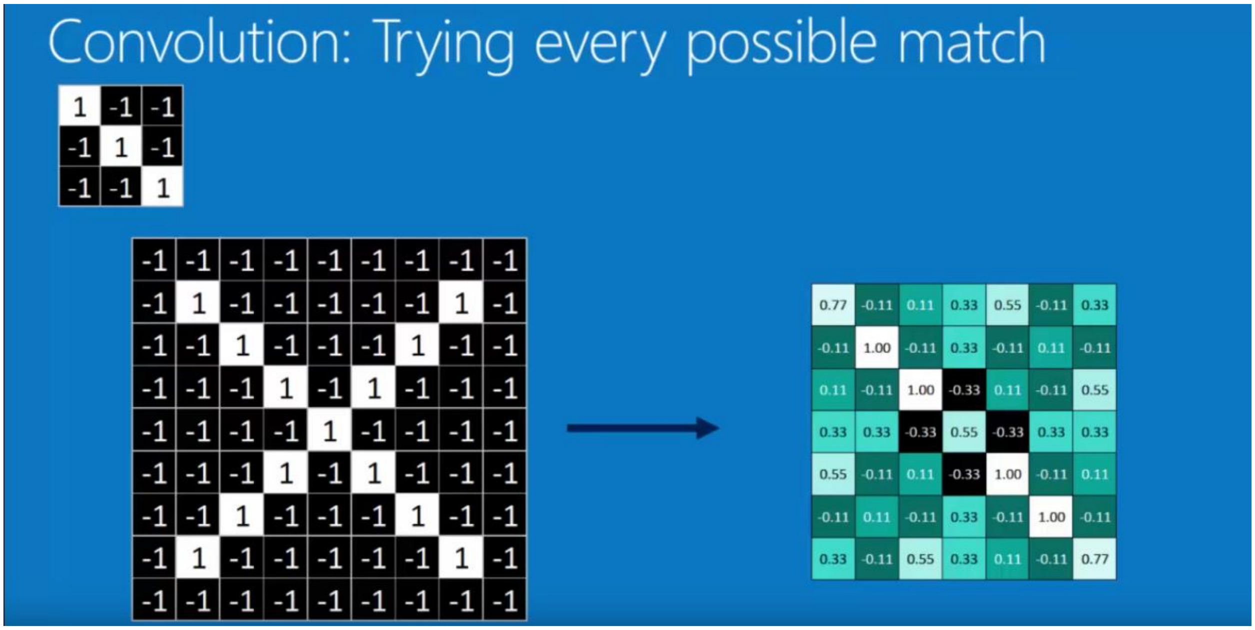


FEATURES MATCH THE PIECES OF THE IMAGE

Steps involved are as follows:

1) Line up the feature and the image patch
2) Multiply each feature pixel with corresponding feature pixel
3) Add them up,
4) Divide by total number of pixels in the feature

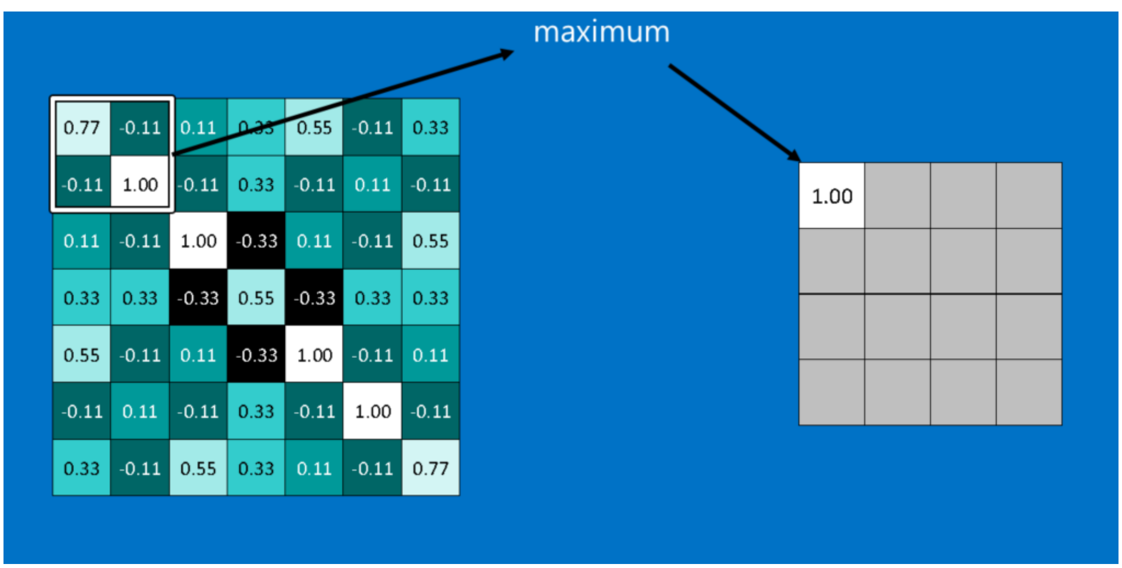
1st Trick Convolution (self-explanatory → divide & check parts of the image for possible match)



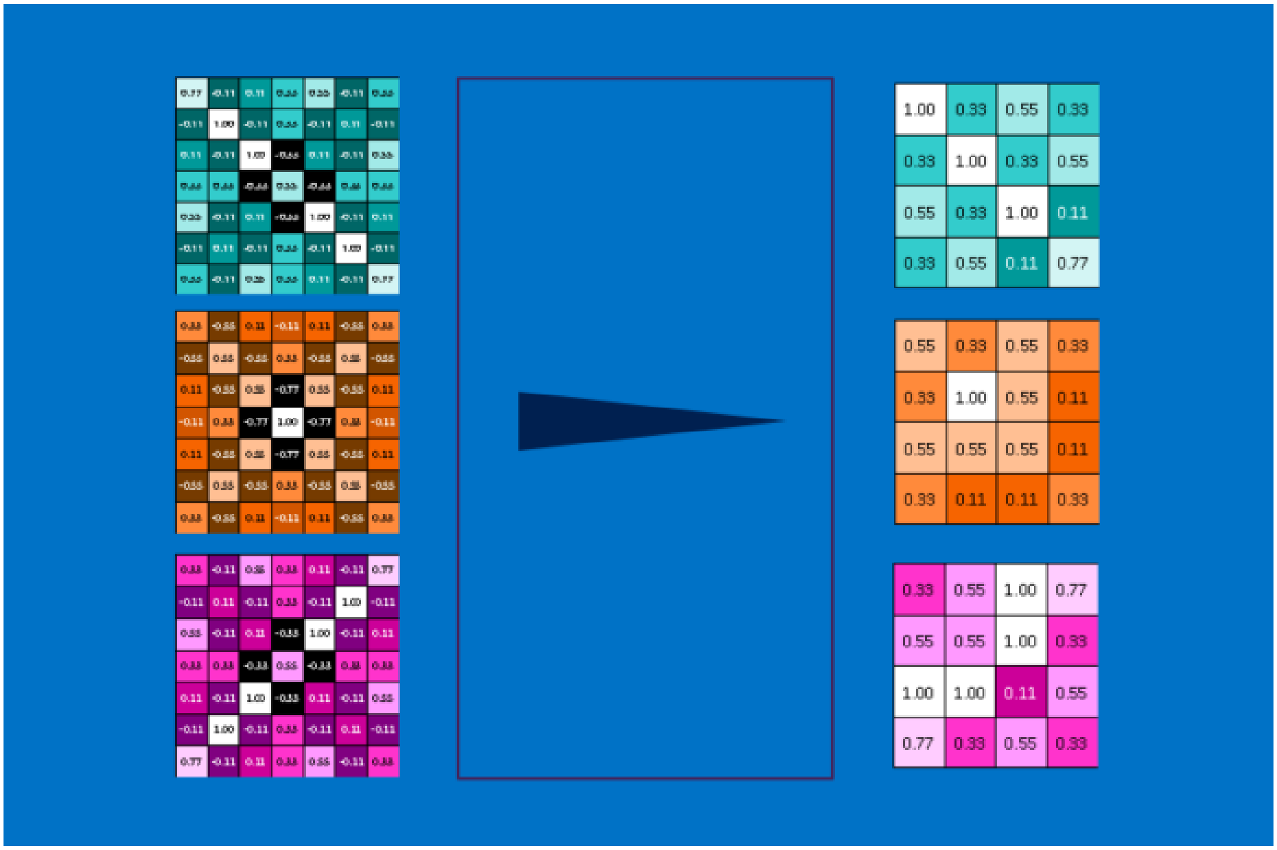
2nd Trick Pooling (Reduce the image dimensionality)

Another tool that CNNs use is called pooling. Pooling is a way to take large images and shrink them down while preserving the most important information in them.

It consists of stepping a small window across an image and taking the maximum value from the window at each step. In practice, a window 2 or 3 pixels on a side and steps of 2 pixels work well

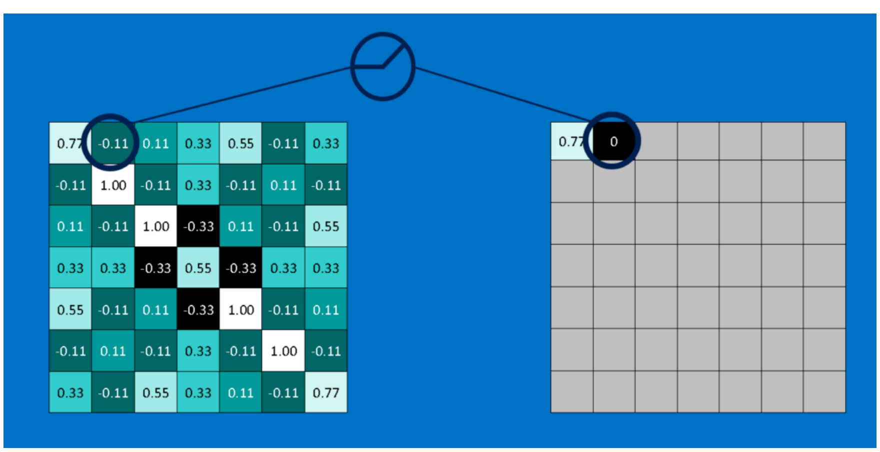


After pooling, an image has about a quarter as many pixels as it started with. Because it keeps the maximum value from each window, it preserves the best fits of each feature within the window. This means that it doesn't care so much exactly where the feature fit as long as it fit somewhere within the window. The result of this is that CNNs can find whether a feature is in an image without worrying about where it is. This helps solve the problem of computers being way too literal



3rd Normalization using **Rectified Linear Units**

For every negative integer, swap it out with 0. This helps the CNN stay mathematically healthy by keeping learned values away from near 0 or going toward infinity



The deep learning then involves putting these layers together in order: Convolution → ReLU → Pooling → Fully connected

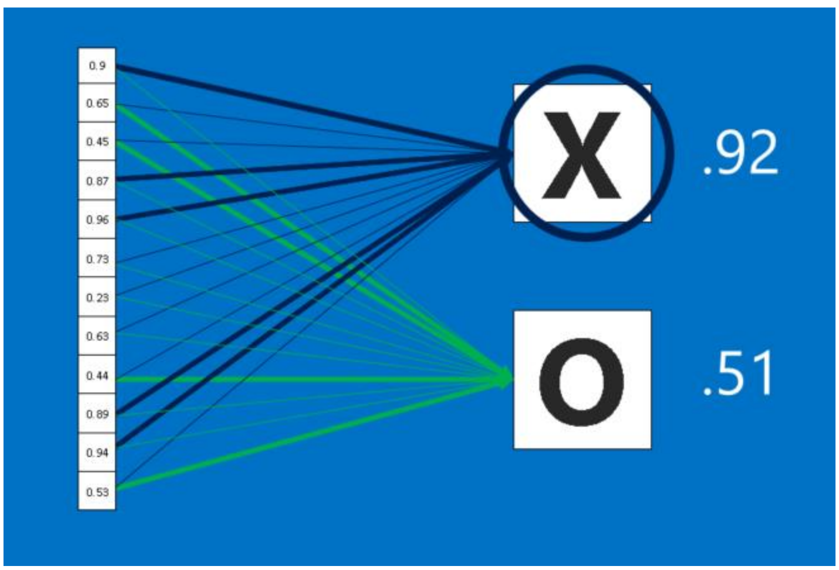
What is the last step (fully connected?)

Fully connected layers take the high-level filtered images and translate them into votes. In our case, we only have to decide between two categories, X and O.

Fully connected layers are the primary building block of traditional neural networks. Instead of treating inputs as a two-dimensional array, they are treated as a single list and all treated identically. Every value gets its own vote on whether the current image is an X or and O.

Some values are much better than others at knowing when the image is an X, and some are particularly good at knowing when the image is an O. These get larger votes than the others. These votes are expressed as weights, or connection strengths, between each value and each category.

Fully connected layers, like the rest, can be stacked because their outputs (a list of votes) look a whole lot like their inputs (a list of values)



IMP. CRITERIA (that are adjusted by modified hyperparameters)

CONVOLUTION

Number of features

Size of features

POOLING

Window size

Window stride

FULLY CONNECTED

Number of neurons

DESIGN CONSIDERATIONS FOR CNN'S

- Architecture
- How many of each type of layer?
- In what order?
- Design new types of layers to add into the web

[CNN's] MORE USES

For instance, audio signals can be chopped into short time chunks, and then each chunk broken up into bass, midrange, treble, or finer frequency bands. This can be represented as a two-dimensional array where each column is a time chunk and each row is a frequency band. "Pixels" in this fake picture that are close together are closely related. CNNs work well on this. This helps creative visualizations