PROFESSIONAL CERTIFICATE IN MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Office Hour #23 with Matilde D'Amelio

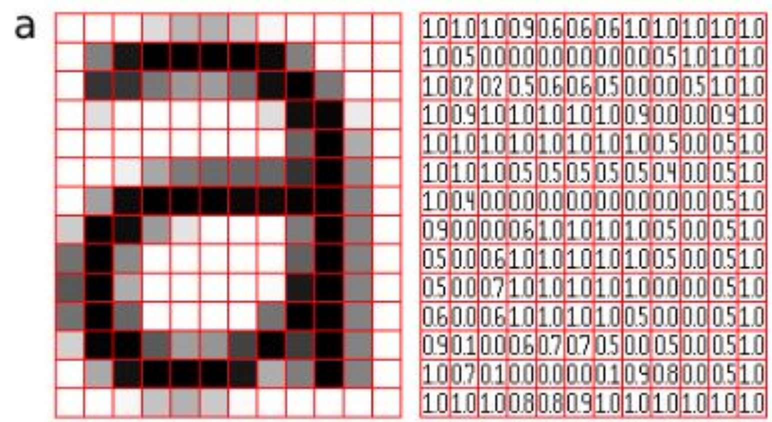
September 1, 2022 at 9 pm UTC

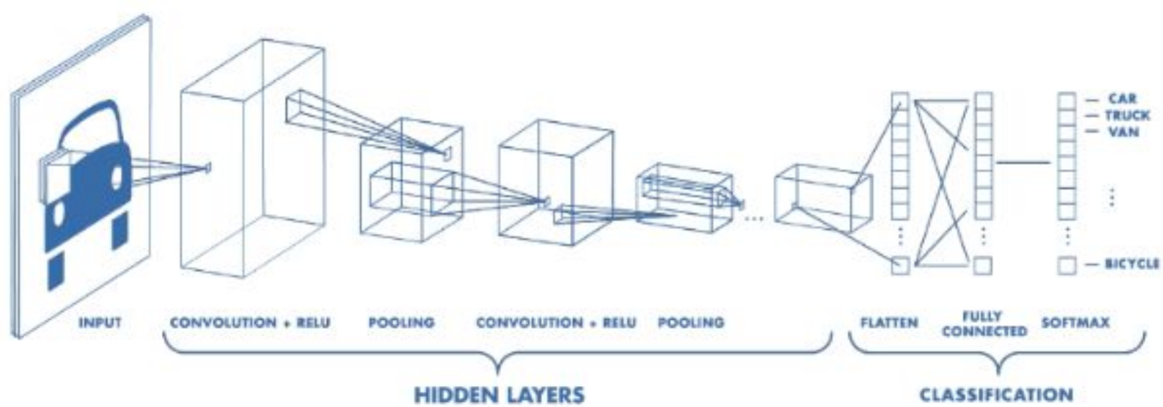
Berkeley Engineering | BerkeleyHaas

Convolutional Neural Network

A Convolutional Neural Network, process data that has a grid-like topology, such as an image. A digital image is a binary representation of visual data. It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright and what color each pixel should be

A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer.





Convolutional Neural Network

Convolution Layer

The convolution layer is the core building block of the CNN. It carries the main portion of the network's computational load.

This layer performs a dot product between two matrices, where one matrix is the set of learnable parameters otherwise known as a **kernel**, and the other matrix is the restricted portion of the receptive field. The kernel is spatially smaller than an image but is more in-depth. This means that, if the image is composed of three (RGB) channels, the kernel height and width will be spatially small, but the depth extends up to all three channels.

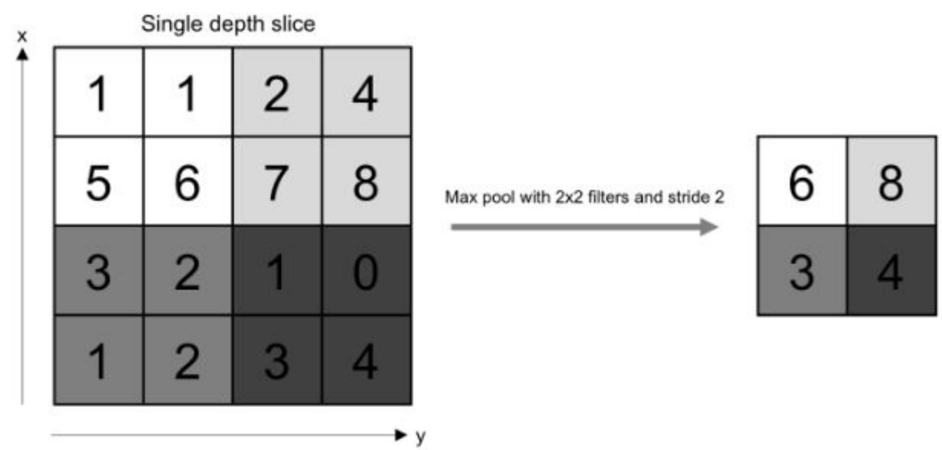
During the forward pass, the kernel slides across the height and width of the image-producing the image representation of that receptive region. This produces a two-dimensional representation of the image known as an activation map that gives the response of the kernel at each spatial position of the image. The sliding size of the kernel is called a stride.

Convolutional Neural Network

Pooling Layer

The pooling layer replaces the output of the network at certain locations by deriving a summary statistic of the nearby outputs. This helps in reducing the spatial size of the representation, which decreases the required amount of computation and weights. The pooling operation is processed on every slice of the representation individually.

There are several pooling functions such as the average of the rectangular neighborhood, L2 norm of the rectangular neighborhood, and a weighted average based on the distance from the central pixel. However, the most popular process is **max pooling**, which reports the maximum output from the neighborhood.



Convolutional Neural Network

Fully Connected Layer

Neurons in this layer have full connectivity with all neurons in the preceding and succeeding layer as seen in regular FCNN. This is why it can be computed as usual by a matrix multiplication followed by a bias effect.

The FC layer helps to map the representation between the input and the output.

Convolutional Neural Network: Application

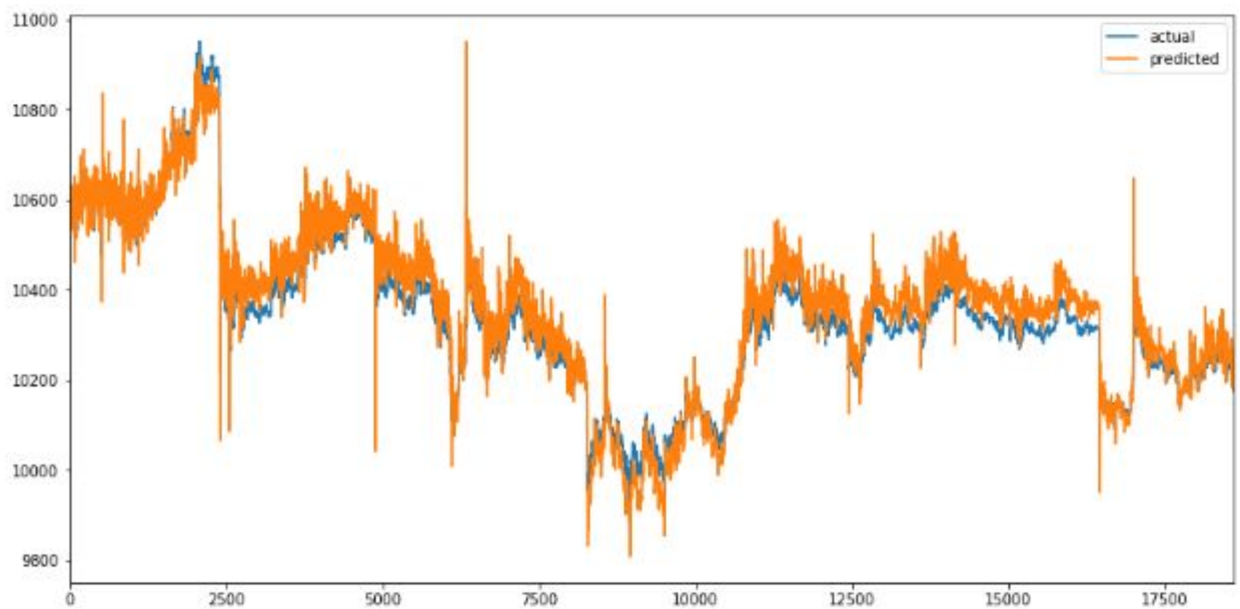
1. **Object detection**: With CNN, we now have sophisticated models like R-CNN, Fast R-CNN, and Faster R-CNN that are the predominant pipeline for many object detection models deployed in autonomous vehicles, facial detection, and more.

2. **Semantic segmentation**: In 2015, a group of researchers from Hong Kong developed a CNN-based Deep Parsing Network to incorporate rich information into an image segmentation model. Researchers from UC Berkeley also built fully convolutional networks that improved upon state-of-the-art semantic segmentation.

3. **Image captioning**: CNNs are used with recurrent neural networks to write captions for images and videos. This can be used for many applications such as activity recognition or describing videos and images for the visually impaired. It has been heavily deployed by YouTube to make sense to the huge number of videos uploaded to the platform on a regular basis.

Long Short Term Memory (for Time Series)

Eg: train a Long Short Term Memory Neural Network (LSTM) on Bitcoin trading data and use it to predict the price of unseen trading data.



Actual and predicted VWAP on the test set

Batch Normalisation

In **deep learning**, preparing a **deep neural network** with many layers as they can be delicate to the underlying **initial random weights** and design of the learning algorithm.

One potential purpose behind this trouble is the distribution of the inputs to layers somewhere down in the network may change after each mini-batch when the weights are refreshed. This can make the learning algorithm always pursue a moving target. This adjustment in the distribution of inputs to layers in the network has alluded to the specialized name **internal covariate shift**.

The challenge is that the model is refreshed layer-by-layer in reverse from the output to the input utilizing an estimate of error that accept the weights in the layers preceding the current layer are fixed.

Batch normalization gives a rich method of parametrizing practically any deep neural network. The reparameterization fundamentally decreases the issue of planning updates across numerous layers.

QUESTIONS?