DEEP LEARNING (P-3)

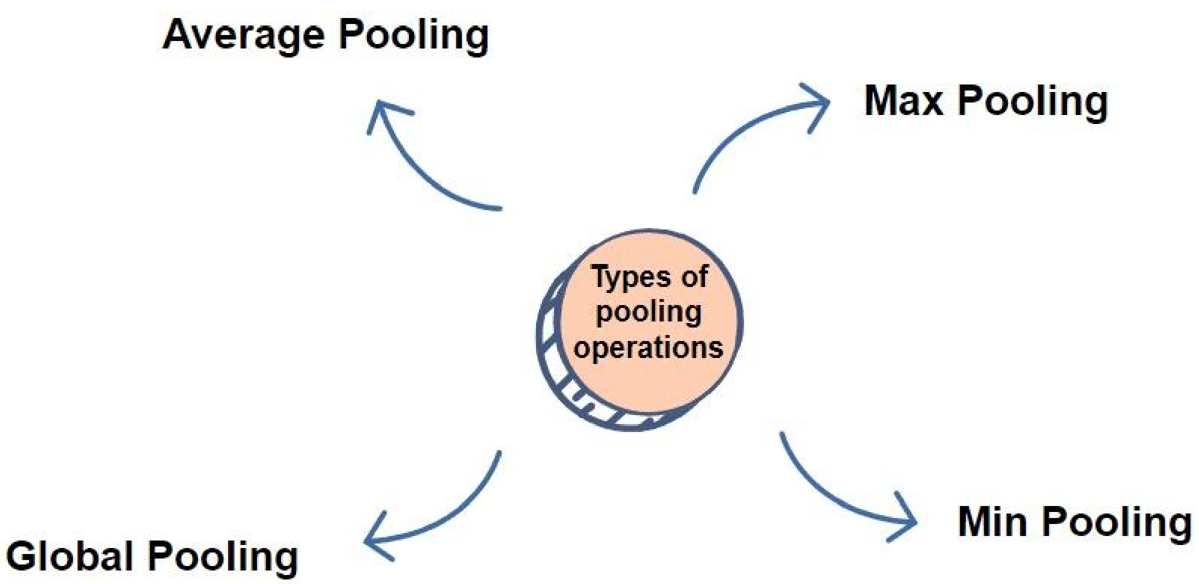
techworldthink • March 08, 2022

6. What is max pooling in the context of CNN?

In a convolutional neural network, pooling layers are applied after the convolutional layer. The main purpose of pooling is to reduce the size of feature maps, which in turn makes computation faster because the number of training parameters is reduced.

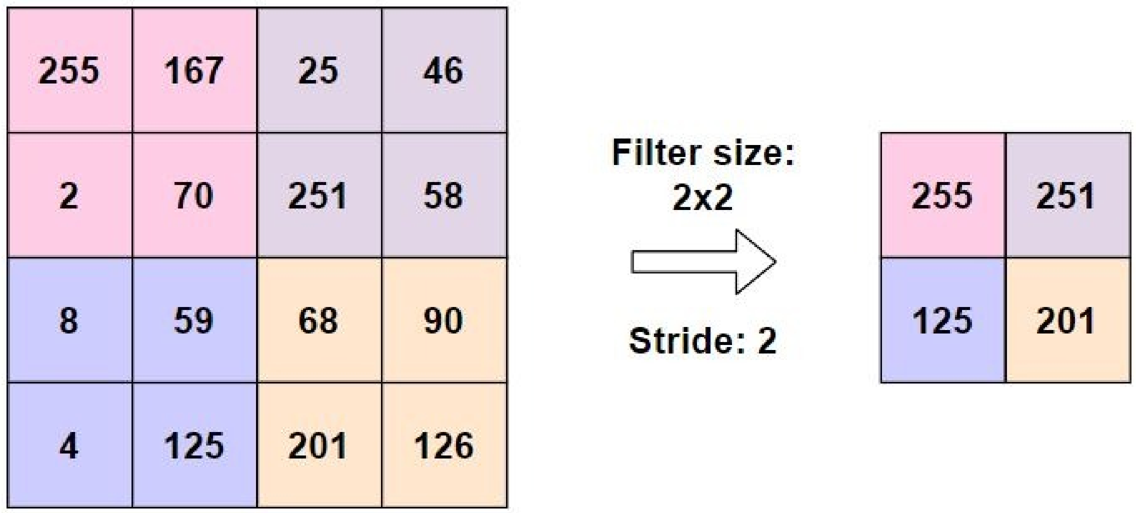
The pooling operation summarizes the features present in a region, the size of which is determined by the pooling filter. If a filter has the dimensions of 2x2, then the region that is summarized is also of the size 2x2.

> *Note: The size of a filter is usually much smaller than the size of the feature map.*



Max Pooling

In this type of pooling, the summary of the features in a region is represented by the maximum value in that region. It is mostly used when the image has a dark background since max pooling will select brighter pixels.Can be implemented using MaxPooling2D in keras.



Min Pooling

In this type of pooling, the summary of the features in a region is represented by the minimum value in that region. It is mostly used when the image has a light background since min pooling will select darker pixels.Can be implemented using MinPooling2D in keras.

Average Pooling

In the third type of pooling, the summary of the features in a region are represented by the average value of that region. Average pooling smooths the harsh edges of a picture and is used when such edges are not important.Can be implemented using AveragePooling2D in keras.

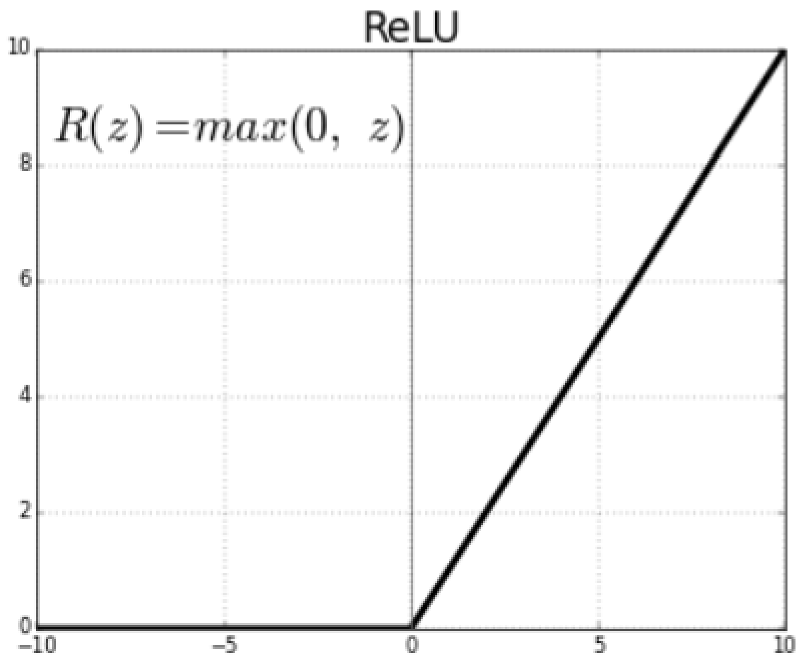
7. Explain ReLU.

In a neural network, the activation function is responsible for transforming the summed weighted input from the node into the activation of the node or output for that input.

The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

- The sigmoid and hyperbolic tangent activation functions cannot be used in networks with many layers due to the vanishing gradient problem.
- The rectified linear activation function overcomes the vanishing gradient problem, allowing models to learn faster and perform better.
- The rectified linear activation is the default activation when developing multilayer Perceptron and convolutional neural networks.

ReLU is the most used activation function in the world right now



Equation : f(x) = max(o,x)

Range : (o to infinity)

Pros:

1. The function and its derivative both are monotonic.
2. Due to its functionailty it does not activate all the neuron at the same time
3. It is efficient and easy for computation.

Cons:

1. The outputs are not zero centered similar to the sigmoid activation function

2. When the gradient hits zero for the negative values, it does not converge towards the minima which will result in a dead neuron while back propagation.

8. Explain the problem of vanishing gradients.

> ***What is Vanishing Gradients?***
>
> *Vanishing Gradient occurs when the derivative or slope will get smaller and smaller as we go backward with every layer during backpropagation.*
>
> *When weights update is very small or exponential small, the training time takes too much longer, and in the worst case, this may completely stop the neural network training.*
>
> *A vanishing Gradient problem occurs with the sigmoid and tanh activation function because the derivatives of the sigmoid and tanh activation functions are between 0 to 0.25 and 0–1. Therefore, the updated weight values are small, and the new weight values are very similar to the old weight values. This leads to Vanishing Gradient problem. We can avoid this problem using the ReLU activation function because the gradient is 0 for negatives and zero input, and 1 for positive input.*

9. Write a note on auto encoders.

Autoencoders are an unsupervised learning technique that we can use to learn efficient data encodings. Basically, autoencoders can learn to map input data to the output data. While doing so, they learn to encode the data. And the output is the compressed representation of the input data. The main aim while training an autoencoder neural network is dimensionality reduction.

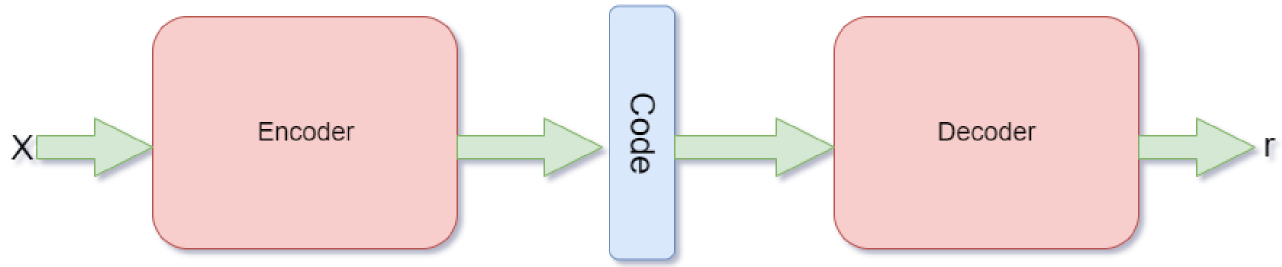
"Autoencoding" is a data compression algorithm where the compression and decompression functions are

1) data-specific,

2) lossy, and

3) learned automatically from examples rather than engineered by a human.

The following image shows the basic working of an autoencoder. It is just a basic representation of the working of the autoencoder.



1.An autoencoder is a feed-forward neural net whose job it is to take an input x and predict x.

2 In another words, autoencoders are neural networks that are trained to copy their inputs to their outputs.

Autoencoders consist of an encoder h = f (x) taking an input x to the hidden representation h and a decoder xˆ = g(x) mapping the hidden representation h to the input xˆ

- An autoencoder is a data compression algorithm.
- A hidden layer describes the code used to represent the input.
- It maps input to output through a compressed representation code.

The latent space is simply **a representation of compressed data in which similar data points** are closer together in space. Latent space is useful for learning data features and for finding simpler representations of data for analysis.

10. Explain the idea behind cross entropy.

Entropy is defined as the randomness or measuring the disorder of the information being processed in Machine Learning.

Cross-entropy is a measure of the difference between two probability distributions for a given random variable or set of events.

You might recall that information quantifies the number of bits required to encode and transmit an event. Lower probability events have more information, higher probability events have less information.

In information theory, we like to describe the "*surprise*" of an event. An event is more surprising the less likely it is, meaning it contains more information.

- Low Probability Event (*surprising*): More information.
- Higher Probability Event (*unsurprising*): Less information.

Information *h(x)* can be calculated for an event *x*, given the probability of the event *P(x)* as follows:

$$h(x) = -\log(P(x))$$