CODE BASICS

DEEP LEARNING TAKEAWAYS

codebasics.io

Chapter:

Convolutional Neural Networks

What is CNN? Convolution, Kernels, Pooling and Beyond

1. Convolutional Neural Networks (CNNs) excel at processing grid-like data such as images, identifying patterns through convolutions.
2. They use kernels (filters) to extract spatial features like edges and textures from input data.
3. Pooling layers reduce spatial dimensions, retaining essential information while minimizing computation and overfitting.
4. CNN architectures combine convolution, pooling, and fully connected layers to learn hierarchical feature representations for tasks like image classification and object detection.
5. CNNs by design can not handle scale and rotation. For this we need to have images in training dataset that contains the variety in scale and roation. In the absence of such images, we can use data augmentation to generate news images from the original dataset

Padding and Strides

1. Padding preserves input dimensions during convolution, ensuring no loss of edge information.
2. Strides control the movement of the convolutional filter, affecting output size and computation speed.
3. Padding techniques like "same" and "valid" balance between maintaining dimensions and reducing output size.
4. Adjusting padding and strides can influence feature extraction granularity and network efficiency.

Data Augmentation

1. Data augmentation increases the diversity of the training dataset by applying transformations like rotation, flipping, and scaling.
2. Enhances model generalization by exposing it to varied scenarios, reducing overfitting.
3. Common techniques include geometric transformations, color adjustments, cropping, and adding noise.
4. Augmentation is performed dynamically during training, ensuring the model sees a new variation in each epoch.
5. Particularly effective in computer vision tasks where gathering more data can be costly or impractical.

What is Transfer Learning

1. Transfer learning leverages pre-trained models to solve new, related tasks with limited data.
2. It significantly reduces training time by reusing learned features from large datasets.
3. Commonly used in tasks like image classification and natural language processing to achieve high accuracy with minimal effort.
4. Transfer learning involves fine-tuning a pre-trained model or using it as a fixed feature extractor.
5. Ideal for scenarios with limited data, enabling effective learning without starting from scratch.