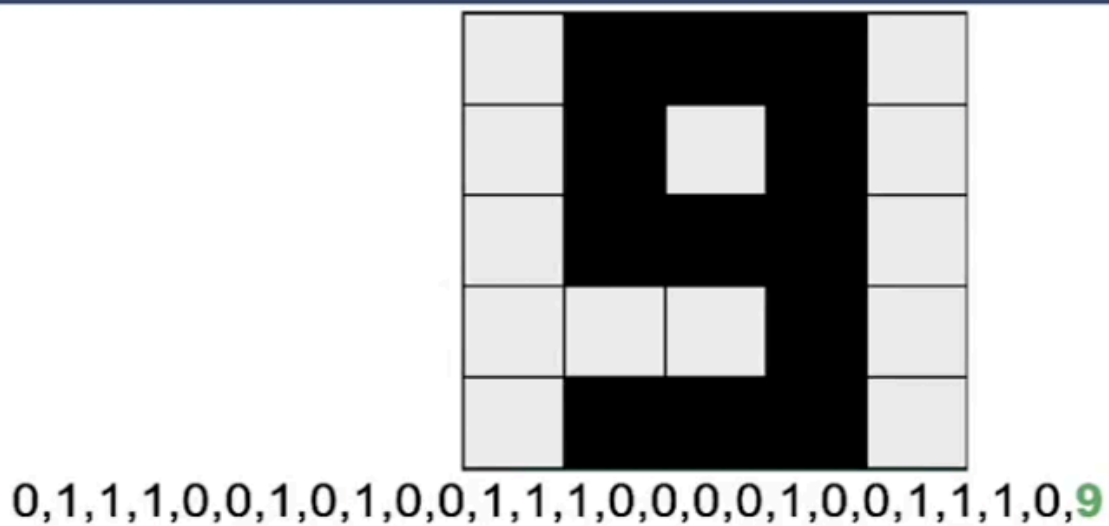


Relatório 15 - Prática: Redes Neurais (II)

Higor Miller Grassi

- **Pixels:** São vários pequenos pontos, chamados de pixels, sendo definido pela quantidade da horizontal x vertical (mutilando, tendo o valor total de pixel), considerado a menor informação disponível em uma imagem, cada um dos pixel é composto por 3 canais que representam as 3 cores básicas (RGB), utilizando elas, conseguimos criar todas as outras cores disponíveis, e ao transformar a imagem em cinza, os números no rgb se tornam iguais, assim, o processamento do algoritmo será mais rápida devido menos informações para ser processadas. Para analisar imagens coloridas, temos que pegar o valor total de pixel e multiplicar por 3 para obtermos os inputs, indica que precisaríamos de x (resultado anterior) de neurônios.

- Estratégia para descobrir números: Ao enviar números, será separado uma imagem de 4x5 pixels e preenchendo o array com 0 onde não está preenchido e 1 onde está preenchido, assim conseguindo descobrir o número.



-Extração de características: nesta abordagem, devemos analisar as imagens para extrair as características únicas de cada exemplo em que se mantém, não há alternâncias no banco de imagens.

- Rede Neural Convolutacional: Não irá utilizar todos os pixels das imagens, economizando e no caso de emojis por exemplo, aos arredores todos são iguais, assim não enviando os pixels desnecessários e realizar um pré processamento para as características principais, e utiliza uma rede neural densa, mas no início faz a transformação dos dados escolhendo os pixels que diferencia uma da outra com os passos principais de: Operador de Convolução, Pooling, Flattening e Rede Neural Densa.

-Operador de Convolução: Com esta imagem 7x7(49 pixels), realizamos a conta com o detector de características(kernel/filtro) multiplicando o pixel da mesma posição, tendo o resultado 0 como no exemplo, na sequência e avançando a parte amarela da esquerda para a direita e depois de cima para baixo, realizando o mesmo cálculo e assim preenchendo a matriz do mapa de características, com o valor negativo transformado em 0. Sendo gerado um conjunto de mapas de características e a rede neural decidirá qual será mais eficiente

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	1	1
0	1	0	1	1	0	0
0	1	0	1	1	0	1
0	1	0	0	0	1	1

Imagem

x

1	0	0
1	0	1
0	1	1

Detector de características

=

0				

Mapa de características

$$0 * 1 + 0 * 0 + 0 * 0 + 0 * 1 + 1 * 0 + 0 * 1 + 0 * 0 + 0 * 1 + 0 * 1 = 0$$

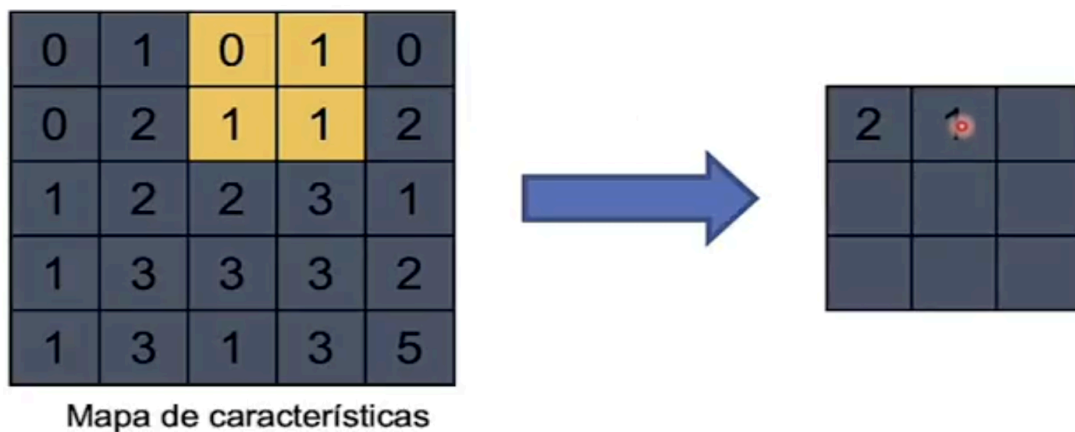
É obtido este mapa de características de 5x5(25 pixels), que seria o resultado do filtro aplicado anteriormente(pixels mais importantes), tendo resultados diferentes em cada detector de características diferentes, como a de identidade, ridge(bordas), sharpen(aumenta nitidez), box blur(efeito de serrilhado), gaussian blur, unsharp masking(parecido com o sharpen).

0	1	0	1	0
0	2	1	1	2
1	2	2	3	1
1	3	3	3	2
1	3	1	3	5

Mapa de características

- **Pooling:** Diminuir ainda mais a dimensionalidade dos dados e para que uma rede neural consiga fazer a detecção em ambientes diferentes, possibilitando a rede neural se adaptar e possíveis mudanças.

Dada a mapa de características, o cálculo do max pooling, para cada mapas de características, é feito pegando o maior valor, avançando de mesma forma que o operador de convolução, da esquerda para a direita e de cima para baixo, assim conseguindo enfatizar as características principais



- **Flattening:** Após realizar o max pooling, é pego a matriz e transformada em um vetor, e cada um desses valores é enviado para uma camada de entrada de uma rede neural densa, e todos os cálculos serão executados.

-**Rede Neural Densa:** Tendo a camada de saída como anteriormente visto, para cada classes, encontrando o padrão de por exemplo 3 classes sendo representadas por 3 números (1,3,9), tendo a 5 camada oculta, no caso quando houver um valor maior nas 2 primeiras, sera o numero 1, quando houver as duas últimas camadas ocultas o maior valor, a probabilidade de ser o 3 é maior, e quando a camada oculta do meio tiver um valor maior, quando tratar do número 9.

- **Augmentação de Dados:** Serve para aumentar a diversidade do seu conjunto de dados de treinamento, o que ajuda a evitar o overfitting e melhora a generalização do modelo. No contexto de redes neurais convolucionais para tarefas de imagem, a argumentação permite aplicar transformações aleatórias nas imagens durante o treinamento, criando diferentes variações das imagens originais.

Conclusão: Neste card, aprendi como fazer uma rede neural usando imagens como dados, aprendendo os desde os conceitos básicos como pixels ate as técnicas utilizadas para tal ato, como os pilares: Operador de Convolução, Pooling, Flattening e Rede Neural Densa. Este aprendizado permitiu obter uma compreensão mais profunda sobre o funcionamento das redes neurais para imagens, além de me capacitar a implementar e treinar essas redes para solucionar problemas do mundo real de maneira eficiente.