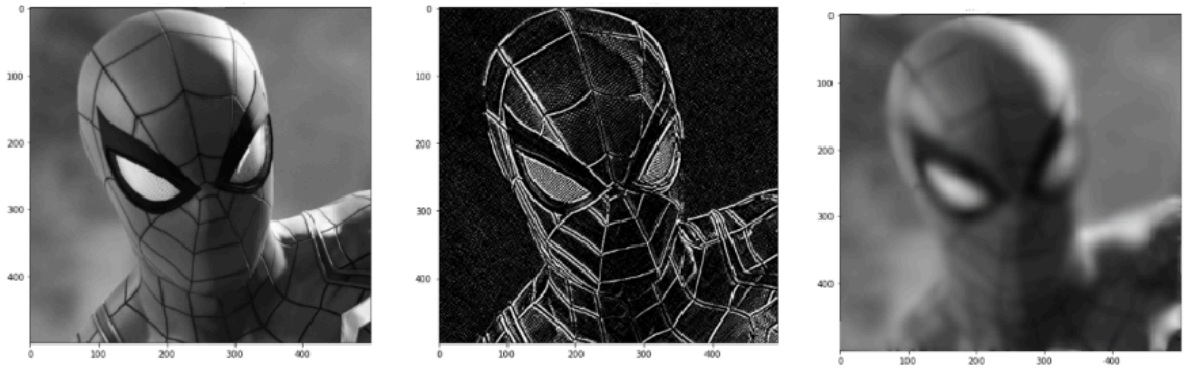
Gabriel Duarte Santos

Relatório - Card 15 - Redes Neurais Convolucionais II

Convolutional Neural Networks

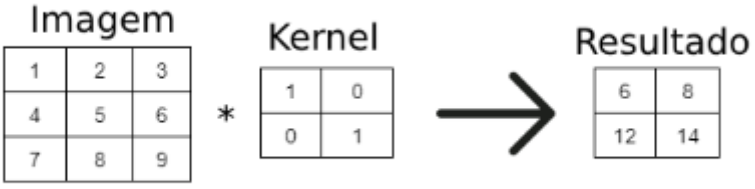
A convolução é uma operação central em processamento de sinais e visão computacional, existindo apenas dois requisitos: adicionar e multiplicar. A primeira perspectiva que se tem da convolução é ser um modificador de imagem, aplicando um blur ou um detector de bordas, como feito no card anterior.



Mecânicas da Convolução

- Multiplicação Elemento a Elemento: O filtro (kernel) desliza sobre a imagem de entrada, multiplicando cada elemento do filtro pelos pixels correspondentes da imagem.
- Soma dos Produtos: Os produtos resultantes são somados para formar um valor no feature map.
- Deslocamento do Filtro: O filtro é deslocado sobre a imagem e o processo se repete

Usando como exemplo se a imagem original do homem-aranha fosse uma matriz 3x3, e que fossemos aplicar um kernel 2x2. Aplicamos a convolução:



Com o kernel na primeira posição no canto superior esquerdo temos:

$$(1 * 1) + (0 * 2) + (0 * 4) + (1 * 5) = 6$$

Na segunda posição, indo a direita:

$$(1 * 2) + (0 * 3) + (0 * 5) +* 1 * 6) = 8$$

Na terceira posição, descendo e voltando a esquerda:

$$(1 * 4) + (0 * 5) + (0 * 7) + (1 * 8) = 12$$

E por fim, na quarta posição, indo novamente para a direita:

$$(1 * 5) + (0 * 6) + (0 * 8) + (1 * 9) = 14$$

Modos de Convolução

Existem diferentes modos de aplicar a convolução que afetam o tamanho da saída:

- Valid: A saída é de tamanho $N - K + 1$, onde N é o tamanho da entrada e K é o tamanho do kernel.
- Same: A saída tem o mesmo tamanho que a entrada, N. Isso é conseguido adicionando preenchimento à imagem.
- Full: A saída é de tamanho $N + K - 1$. Este modo é menos comum, resultando em uma saída maior que a entrada

Mode	Output Size	Usage
Valid	N - K + 1	Typical
Same	N	Typical
Full	N + K - 1	Atypical

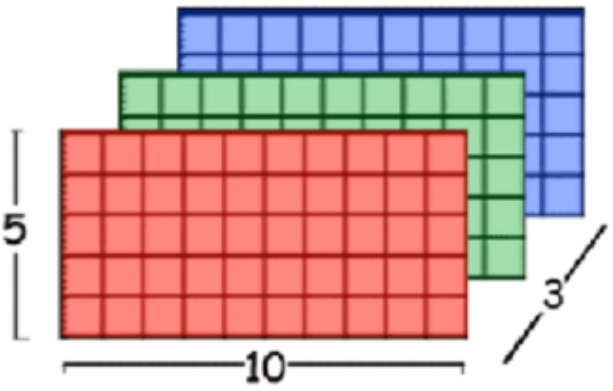
A aplicação de filtros nas imagens é essencial para a descoberta de padrões. Eles ajudam a identificar e destacar as características mais importantes de uma imagem, diferenciando os elementos que se desviam do padrão e facilitando o reconhecimento e a análise.

Por Que os Índices Começam em 0 na Programação?

Um dos principais motivos para iniciar a indexação em 0 é a facilidade que isso traz para o cálculo de deslocamento e endereços de memória. Começando o índice em 0 facilita a compreensão e a manipulação de estrutura de dados. Por exemplo, aplicando um filtro sobre uma imagem, se o índice começasse em 1, teríamos que lidar com um deslocamento entre o estado inicial (0) e o índice dos elementos (que começou em 1), podendo complicar o entendimento do código.

Convolução em Imagens Coloridas

Para o caso das imagens em preto e branco tínhamos sempre imagens em 2D, onde as dimensões são altura e largura. Porém, em imagens coloridas, a representação se torna 3D, pois cada pixel é composto por três valores, correspondendo às intensidades das cores RGB.



Para aplicar a convolução em imagens coloridas é necessário o uso de múltiplos filtros. Cada filtro é aplicado separadamente a cada uma das camadas, obtendo um conjunto de imagens em 2D. Esse conjunto de imagens então é empilhado, formando uma representação final também 3D. Assim, o resultado da convolução de uma imagem colorida é uma camada tridimensional, onde a profundidade é igual ao número de filtros utilizados.

Esse conjunto de imagens funciona como um “mapa” final da imagem, onde as características relevantes foram destacadas pelos filtros. Esse “mapa” guia a rede neural para classificar os elementos importantes na imagem. Uma grande vantagem dos filtros de convolução é que eles são aplicados de forma compartilhada em toda a imagem, o que reduz a complexidade computacional e permite que a rede aprenda de maneira mais eficiente.

Arquitetura de uma CNN

Uma CNN típica é composta por dois estágios. O primeiro estágio é uma série de camadas convolucionais, sendo um transformador de recursos que encontra recursos em imagens, como bordas e texturas. Já o segundo estágio é uma série de camadas densas (totalmente conectadas) que realizam a classificação ou regressão com base nas características extraídas.

Pooling

Pooling é uma técnica usada para reduzir a dimensionalidade da imagem, diminuindo o número de parâmetros e o custo computacional. Existem dois tipos principais:

- Max Pooling: Seleciona o valor máximo de uma região da imagem, mantendo as características mais fortes e relevantes. É o mais utilizado porque preserva as características mais significativas.
- Average Pooling: Calcula a média dos valores em uma região, mas é menos comum que o Max Pooling.

O Max Pooling é preferido porque destaca as características mais importantes, ajudando a rede a focar nos elementos mais relevantes da imagem, o que melhora a precisão do modelo.

Ao aplicar uma sequência de camadas de convolução e pooling, ocorre uma redução progressiva do tamanho da imagem, destacando as características essenciais. A convolução é responsável por extrair os padrões mais relevantes, enquanto o pooling reduz a resolução da imagem, concentrando a informação nas áreas mais significativas. Esse processo permite que a rede neural ignore detalhes irrelevantes e foque na identificação dos padrões mais importantes.

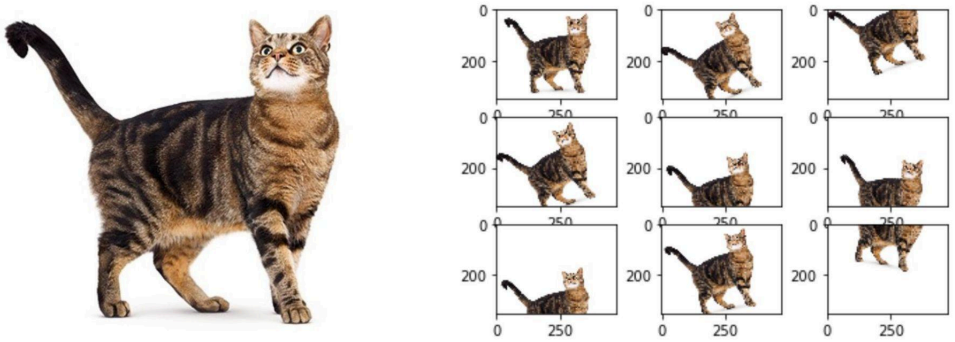
Camadas Densas

No último estágio da arquitetura de uma CNN, as camadas densas (totalmente conectadas) são responsáveis por processar a saída das camadas convolucionais e de pooling. Essas camadas pegam as características extraídas e as utilizam para realizar a tarefa final, como classificação ou regressão.

Data Augmentation

É uma técnica usada para aumentar a quantidade de dados de treinamento ao criar novas versões das imagens existentes, aplicando transformações aleatórias, diversificando o conjunto de treinamento. Algumas das transformações são:

- Rotação: Girar em diferentes ângulos.
- Zoom: Aumentar e diminuir o zoom.
- Translação: Mover a imagem.
- Reflexão (Flip): Inverter horizontalmente ou verticalmente.



Essas transformações são aplicadas em tempo real durante o treinamento do modelo, para evitar a necessidade de armazenar um grande volume de dados no disco. Gerar essas variações ajuda o modelo a generalizar melhor, lidando melhor com diferentes dados de entrada.

Batch Normalization

Batch Normalization tem como objetivo normalizar as saídas de uma camada neural, fazendo com que a distribuição dos dados seja mais estável. Uma técnica muito adotada é inserir Batch Normalization entre as camadas de convolução, o que ajuda a manter o treinamento mais estável e eficiente.

Embeddings

Para representar palavras em modelos de aprendizado de máquina, é essencial usar embeddings, que são representações vetoriais de palavras ou frases. Os embeddings permitem capturar as relações semânticas entre palavras, posicionando palavras semelhantes próximas umas das outras no espaço vetorial. Cada palavra é mapeada para um inteiro, que aponta para a posição correspondente na matriz, que é treinada junto com a CNN para otimizar o desempenho do modelo.

Processo de Tokenização e Indexação

O processo começa com uma lista de strings, onde cada string pode ser uma ou mais frases. O tokenizador converte cada string em uma lista de inteiros, onde cada inteiro corresponde a uma palavra. Esses inteiros são então utilizados para indexar uma matriz de pesos (embedding), representando a palavra de forma vetorial.

```
# senteças de teste
sentences = [
    "I like eggs and ham",
    "I love chocolate and bunnies",
    "I hate onions."
    ]
```

```
In [2]: print(sequences)
[[1, 3, 4, 2, 5], [1, 6, 7, 2, 8], [1, 9, 10]]
```

Convolução na Prática

Convolução na vida real

A convolução é amplamente aplicável, tanto em áudio quanto em imagens. Em músicas, por exemplo, a convolução pode ser usada para adicionar efeitos como eco ou reverb, alterando o som. Em imagens, é usada para aplicar filtros, como efeitos de blur ou bloom.

No deep learning, o objetivo é descobrir quais filtros são melhores para extrair características relevantes dos dados, que embora possam parecer estranhos para nós, são muito significativos para o computador, que os utiliza para reconhecer padrões e tomar decisões.

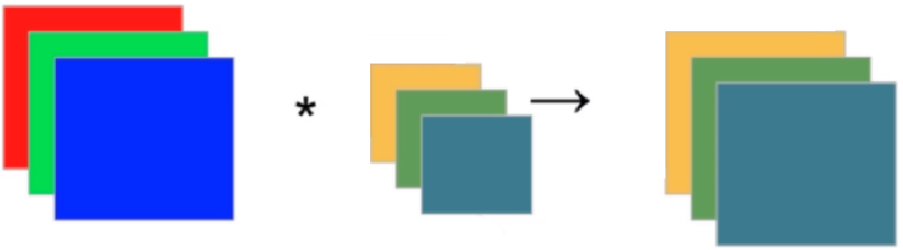
Visões Alternativas da Convolução

A convolução pode ser vista como uma multiplicação de matrizes, onde um filtro é aplicado repetidamente aos dados. Outra perspectiva é a correlação cruzada, que mede a semelhança entre o filtro e os dados, enquanto a convolução se concentra em extrair características. Ambas as técnicas são usadas para o reconhecimento de padrões e análise de sinais e imagens.

Convolutional Neural Networks Description

Convolução em imagens 3D

Além de imagens em 2D, a convolução pode ser aplicada em imagens 3D. Nesse contexto, as imagens têm uma dimensão adicional que pode representar a profundidade, tempo ou outros aspectos, dependendo da aplicação. Para processar essas imagens, utiliza-se filtros de convolução 3D, que se movem nas três dimensões. O resultado é um conjunto de mapas de características tridimensionais, como já mencionado acima.



Formas de Rastreamento em CNN

A identificação dos padrões feitas pelas CNNs são feitos por múltiplas camadas de convolução, tendo cada camada extraindo características específicas.

- Camadas iniciais: Capturam coisas simples, como bordas e linhas.
- Camadas intermediárias: Identificam combinações para formar padrões.
- Camadas finais: Reconhecem características mais específicas, como rostos, animais, etc.

A rede ajusta seus filtros durante o treinamento para capturar melhor as características relevantes das imagens, diminuindo-as até o ponto de grande parte da imagem ser apenas as características necessárias.

Dicas Práticas para o uso de CNN

- Estrutura das Camadas
  - Cada camada de uma CNN processa imagens, transformando as dimensões espaciais (altura e largura) e a profundidade (número de mapas de características).
  - Imagens coloridas têm três canais (RGB), enquanto as saídas das camadas convolucionais têm vários mapas de características.
- Camadas Totalmente Conectadas
  - As camadas totalmente conectadas tipicamente mantêm um número constante de neurônios ou diminuem gradualmente, evitando o overfitting.
  - O número de unidades escondidas por camada deve ser cuidadosamente ajustado, balanceando capacidade e eficiência.
- Importância dos Fundamentos
  - Embora as arquiteturas CNN possam parecer sofisticadas, a maioria delas segue os mesmos princípios fundamentais.

Uso das Loss Functions

São muito importantes para o treinamento de redes neurais, pois medem a diferença entre as previsões do modelo e os valores reais. Existem várias funções de perda, dentre elas:

- Mean Squared Error: É usada para problemas onde a saída é um valor contínuo. Calcula a média dos quadrados das diferenças entre os valores reais e as previsões.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- Binary Cross Entropy: Usada para problemas de classificação binária. Mede a divergência entre as previsões e os valores reais binários (0 ou 1).

$$-\frac{1}{N}\sum_{i=1}^{N}\{y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)\}$$

- Categorical Cross Entropy: Usada para problemas de multiclasse. É uma extensão da Binary Cross Entropy para múltiplas classes, sendo cada classe representada por um vetor one-hot (1 para a classe correta e 0 para outras).

$$-\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\log \hat{y}_{ik}$$

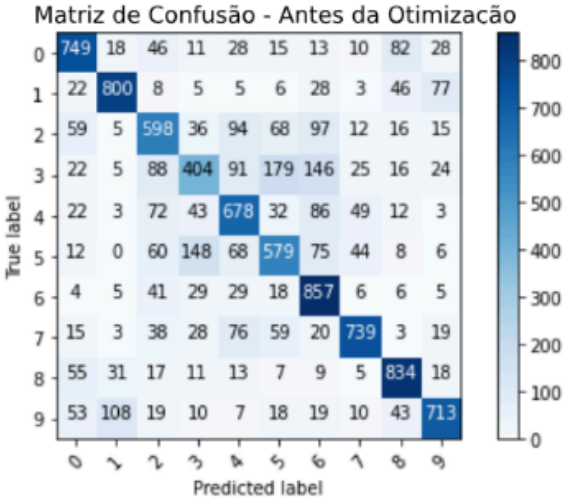
Uso da Descida do Gradiente

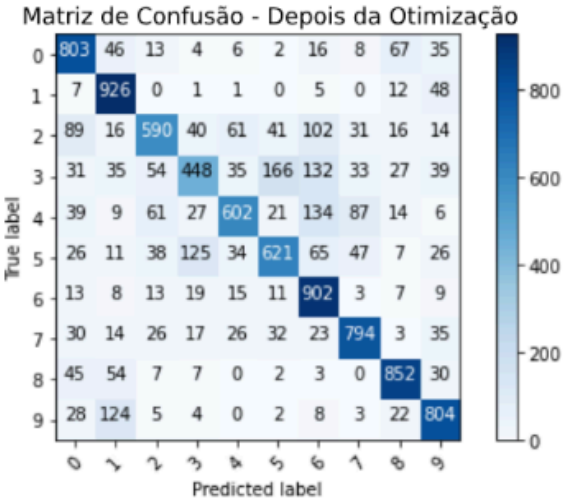
A descida do gradiente é fundamental para o treinamento de redes neurais, pois ajusta os pesos do modelo para minimizar a função de perda. Existem várias variantes da descida do gradiente:

- **Stochastic Gradient Descent:** Em vez de usar todo o conjunto de dados para calcular o gradiente, o SGD utiliza um único exemplo ou uma pequena quantidade de exemplos para cada atualização, o que pode acelerar o processo de treinamento.
- **Momentum:** Esta técnica acelera o SGD ao adicionar um termo que acumula o gradiente das iterações passadas, ajudando a superar barreiras e a manter o progresso em direção a mínimos globais.
- **Variable and Adaptive Learning Rates:** Técnicas como o decaimento da taxa de aprendizado e métodos adaptativos ajustam a taxa de aprendizado durante o treinamento, melhorando a eficiência e a precisão do modelo.
- **Adam:** Adam adapta a taxa de aprendizado para cada parâmetro, utilizando médias móveis dos gradientes e do quadrado dos gradientes, resultando em uma convergência mais rápida e eficiente.

Atividade Prática - Otimizando a CNN CIFAR-10

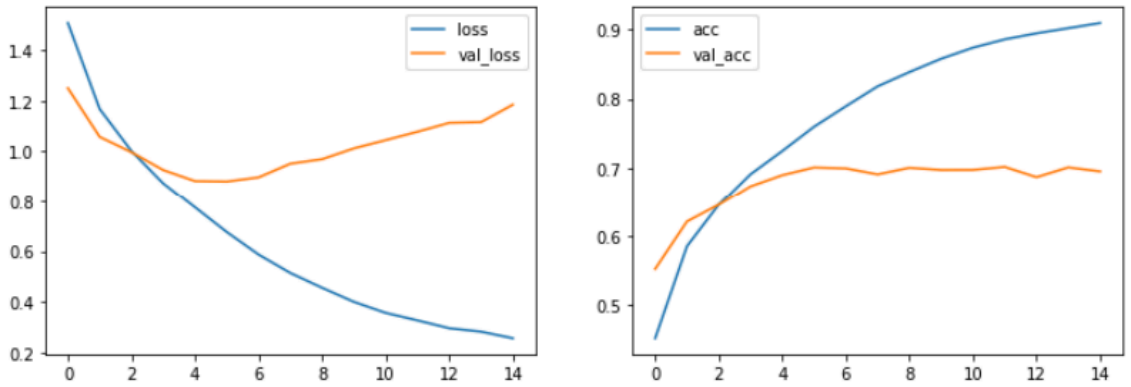
Como parte do estudo sobre CNNs e com os códigos desenvolvidos nas aulas, desenvolvi uma otimização para o código do CIFAR-10, que classifica imagens utilizando uma base de dados com 60.000 imagens de 32x32 pixels. Com essa otimização, obtive uma melhora na precisão e na eficiência do reconhecimento de padrões.



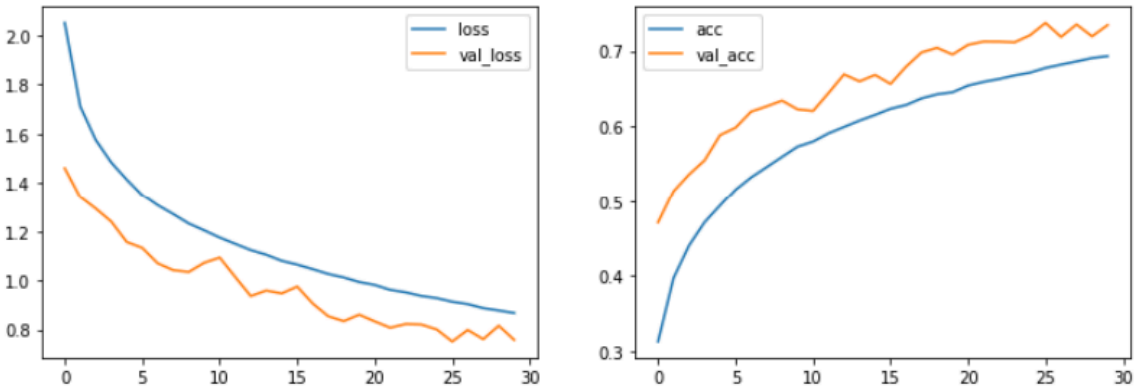


O modelo consiste em várias camadas convolucionais seguidas por camadas densas, com o objetivo de extrair características das imagens. Comparando com o modelo original, aumentei o número de épocas de treinamento, tendo mais tempo de aprendizado. Utilizando do Image Data Generator e aplicando Batch Normalization após cada camada, ocorreu uma melhora na velocidade de treinamento. Adicionando camadas convolucionais extras e ajustando parâmetros como a taxa de dropout, generalizando mais o modelo.
As melhorias resultaram em uma melhor acurácia e uma matriz de confusão mais equilibrada, obtendo mais precisão em várias classes.

Acurácia e Perda por Iteração - Antes da Otimização



Acurácia e Perda por Iteração - Depois da Otimização



Conclusão

As Convolutional Neural Networks (CNNs) são uma inovação crucial no aprendizado de máquina, especialmente para processamento e reconhecimento de imagens. Utilizando suas camadas convolucionais, de pooling e totalmente conectadas, as CNNs são capazes de extrair características de imagens de maneira eficiente, permitindo uma análise detalhada e precisa.
Neste relatório, foi discutido sobre as arquiteturas de CNNs, como VGG, e a importância das funções de perda e técnicas de descida do gradiente no treinamento de modelos. Que são coisas essenciais para desenvolver redes neurais eficazes e robustas, capazes de desempenhar uma variedade de tarefas no campo da inteligência artificial.