

## Relatório 20 - Prática: Visão Computacional (III)

Lucas Augusto Nunes de Barros

### Descrição das atividades

O PyTorch é uma biblioteca de aprendizado de máquina de código aberto criada e mantida pela Meta AI, ganhou popularidade por ser bastante flexível, facilitando o uso e oferecendo suporte à computação com uso de GPU's, acelerando significativamente o treinamento de modelos mais complexos.

No núcleo do PyTorch estão os tensores, estruturas de dados multidimensionais semelhantes aos arrays NumPy. O tensor é como uma generalização de vetores e matrizes para dimensões potencialmente mais altas. Assim como um vetor é um array unidimensional de números e uma matriz é um array bidimensional, um tensor pode ser um array n-dimensional. Essa estrutura flexível é o bloco de construção básico para representar e manipular dados em PyTorch, desde entradas de modelos de machine learning até os pesos e gradientes calculados durante o treinamento.

As operações com tensores no PyTorch são extremamente importantes para manipulação e transformação de dados, que são a matéria prima para qualquer modelo de IA, simples ou complexo, por isso o tratamento de dados é tão fundamental para qualquer aplicação envolvendo redes neurais.

O uso de tensores no contexto de inteligência artificial e treinamento de modelos oferece vantagens cruciais. Sua estrutura multidimensional permite representar de forma natural dados complexos como imagens, sequências de texto e dados tabulares de qualquer dimensão. A otimização dessas operações acelera significativamente os cálculos necessários para o treinamento. Além disso, a capacidade de rastrear automaticamente os gradientes das operações em tensores simplifica enormemente a implementação do algoritmo de *backpropagation*, essencial para o aprendizado de modelos de redes neurais profundas.

A construção de redes neurais com PyTorch envolve a definição da arquitetura através da combinação de camadas predefinidas do módulo *torch.nn*. Essas camadas são organizadas sequencialmente ou em estruturas mais elaboradas dentro de classes, definindo o fluxo de dados através do modelo. A flexibilidade do PyTorch permite a prototipagem rápida e a implementação simplificada de modelos de aprendizado de máquina.

Em aplicações de visão computacional, redes neurais são empregadas para extrair características visuais significativas de imagens. A entrada de uma imagem é agora representada como um tensor multi dimensional, que é então processado por camadas da rede para identificar padrões. A capacidade das redes neurais de aprender representações abstratas diretamente dos dados brutos as tornam ideais para tarefas de detecção e classificação de imagens.

As redes neurais convolucionais (CNNs) profundas são arquiteturas especializadas para o processamento de imagens, utilizando camadas convolucionais para aprender características. O uso de modelos profundos, com múltiplas camadas convolucionais e de *pooling*, permite a captura de padrões cada vez mais complexos

de características visuais. Uma técnica bem utilizada para melhorar o desempenho do treinamento de CNN, principalmente em casos de dataset com baixa variabilidade é a *data augmentation*, que aplica transformações aleatórias nas imagens de treinamento, aumentando a diversidade do conjunto de dados, o que consequentemente melhora a capacidade de generalização do modelo.

Um outro conceito abordado é o de *Autoencoders*, que são redes neurais projetadas para aprender representações compactas e eficientes de dados de forma não supervisionada. Sua arquitetura básica consiste em um codificador (*encoder*) que mapeia a entrada para uma representação de menor dimensão (o espaço latente) e um decodificador (*decoder*) que tem como papel reconstruir a entrada original a partir dessa representação latente. O objetivo do treinamento é minimizar o erro de reconstrução, forçando a rede a capturar as características mais importantes dos dados na representação intermediária. Essa capacidade de aprender representações significativas sem rótulos torna os *autoencoders* úteis para redução de dimensionalidade, remoção de ruído e detecção de anomalias.

Esse conceito também pode ser implementado integrado em redes CNN, onde a CNN atua como o *encoder* extraindo características espaciais da imagem de entrada, enquanto o *decoder* do modelo fica a cargo da reconstrução da imagem com base na representação do latente.

*Autoencoders* variacionais (VAEs) introduzem uma abordagem probabilística à arquitetura tradicional, aprendendo uma distribuição de probabilidade sobre o espaço latente em vez de uma representação determinística. O codificador do VAE gera parâmetros de uma distribuição e uma amostra dessa distribuição é usada pelo decodificador para gerar a saída. Essa formulação permite a geração de novas amostras plausíveis a partir do espaço latente aprendido, tornando os VAEs particularmente úteis para tarefas generativas em visão computacional, como a criação de novas imagens com características semelhantes aos dados de treinamento.

Por fim, a última seção do curso propõe três projetos populares na área, incluindo o *Neural Style Transfer*, que consiste em transferir o estilo artístico de uma imagem para o conteúdo de outra utilizando redes neurais convolucionais. O projeto de *Deep Fake* que gera imagens que manipulam ou substituem a aparência de uma pessoa. Finalmente, o *Image Colorization* se propõe a adicionar cores em imagens em escala de cinza, empregando redes neurais para aprender a relação entre os pixels e as possíveis informações cromáticas.

## Conclusão

A exploração das redes neurais, desde as convolucionais (CNNs) para o processamento de imagens até as arquiteturas de aprendizado não supervisionado como *autoencoders* e *autoencoders* variacionais (VAEs), demonstra a versatilidade e o poder do aprendizado profundo. Projetos práticos como *Neural Style Transfer* e *Image Colorization* ilustram as possibilidades que essas técnicas abrangem para resolver problemas, evidenciando o potencial deste ramo da inteligência artificial.

## **Referências**

[1] Card 20 - The Ultimate Computer Vision and Deep Learning Course

[2] MEDIUM.COM. Neural Style Transfer (NST). Disponível em:  
<https://medium.com/@ferlatti.aldo/neural-style-transfer-nst-theory-and-implementation-c26728cf969d> , acessado em 12 de maio de 2025

[3] KAGGLE.COM. PyTorch-Tutorial (The Classification) . Disponível em:  
<<https://www.kaggle.com/code/basu369victor/pytorch-tutorial-the-classification>> , acessado em 13 de maio de 2025