

## Seção 2 Fundamentos do Deep Learning

Aprendizagem de máquina que usa redes neurais artificiais para modelar e resolver problemas complexos, revolucionando várias áreas na indústria.

**Pytorch:** Biblioteca de aprendizado de máquina de código aberto, usada em desenvolvedores em áreas distintas, como saúde, finanças entre outros. Muito utilizado devido a fácil compreensão, inteirando de forma rápida e sendo excelente para modelos complexos.

**Tensor:** Usada para construir e treinar redes neurais, podendo ser manipulada usando várias operações integradas, conseguindo interpretar dimensões e realizar previsões do modelo dado. São otimizados para cálculos de aprendizado profundo e para gradientes para propagação reversa, podem realizar cálculos em paralelo e uma API intuitiva.

## Seção 3: Construindo redes neurais com Pytorch

**Redes Neurais:** É Um modelo de aprendizado de máquina que são feitos a partir da estrutura do cérebro usada para resolver problemas complexos, funcionando a princípio em 3 camadas, a camada de entrada, camada oculta(onde há os cálculos) e a camada de saída cada uma ligada entre si igual neurônios produzindo uma camada de saída de acordo com pesos. Usada para realizar previsões em dados novos e não vistos.

**Treinar Redes Neurais:** Para realizar o treinamento, temos os pilares principais: definir o objetivo do projeto(classificação de imagens), conjunto de dados(usado para a rede neural identificar padrões nos dados de entrada e dar a saída), pré processamento de dados(normalizar/dimensionar), atualizar os pesos da rede(para chegar mais perto do accuracy perfeito) e realizar o teste do modelo com dados para ver o quão bem ele se generaliza para dados novos.

**Arquitetura do Modelo:** Layout dos nós e camadas individuais na rede neural, usado para definir a estrutura da rede e a sua capacidade de processar os dados, consistindo em uma série de camadas convolucionais(CNN) ou a rede neural recorrente(RNN) usado para processar dados sequências, como texto.

**Funções de Ativação:** Função matemática aplicada na entrada no neurônio para determinar a saída, introduzindo uma não linearidade permitindo a rede neural aprender padrões mais complexos. Como exemplo, uma função usada

frequentemente é a sigmoid(0 a 1), também a função reLU que mapeia qualquer valor de entrada menor que 0 para 0 e qualquer valor maior que 0 para si mesmo.

**Função de Perda:** Usada para medir o desempenho da rede comparando a saída prevista com a saída real da rede neural, retornando um valor do quão longe foi a previsão. Como exemplo, a função MSE que calcula a diferença quadrática média entre eles.

**Otimizadores:** Utilizado para a manipulação dos pesos da rede neural para minimizar a perda durante o treinamento, calculando o gradiente da função de perda em relação aos parâmetros da rede neural e então ajustando os pesos. Como por exemplo o otimizador Adam sendo ideal para trabalhar com grandes volumes de dados e redes neurais complexas.

**Dataset/Dataloader:** Conjunto de dados usado para treinar a rede neural podendo ser imagens, texto ou dados números, já o dataloader ajuda a carregar com eficiência os dados do conjunto e preparar para o treinamento. Quanto maior o dataset mais preciso será, porém irá demorar mais tempo para ser treinado.

## **Seção 4: Redes Neurais para imagens**

**Classificação de imagem:** Pega uma imagem como entrada e produz uma distribuição de probabilidade sobre um conjunto de classes como saída, dentro disso, a classe mais provável é escolhida como rótulo previsto para a imagem. A cnn é usada normalmente para classificação de imagem, consistindo em camadas convolucionais, camadas de resfriamento e camadas totalmente conectadas, usando filtros para extrair características da entrada, realizando cálculos em pixels correspondentes à imagem. Na camada de saída ela sofre uma transformação linear alimentado em uma função softmax para produzir a distribuição de probabilidade das classes, usando uma função de perda e um otimizador durante o treinamento.

**Hyper Parâmetros:** São os parâmetros da rede neural que são definidos antes do treinamento, como a taxa de aprendizado, o número de camadas e o número de neurônios em cada camada. Eles não conseguem ser aprendidos no treinamento, mas influencia diretamente como o modelo aprende com os dados, um exemplo é a taxa de aprendizado, influenciando nos pesos e como são atualizados a cada epoch.

**Normalização de Dados:** Etapa do pré-processamento de dados, normalizar imagens significa ajustar seus valores de pixel para um intervalo padrão, normalmente entre 0 e 1, ajudando a visão computacional a melhorar a sua precisão removendo inconsistências nos dados. Ajustando o brilho das imagens para que não seja tendenciosa para uma imagem clara ou escura.

## Seção 5: Redes Neurais Convolucionais

**CNN(Convolutional Neural Network):** Aprendem a reconhecer padrões e imagens observando pequenas partes da imagem de cada vez, se tornando mais eficiente e precisa do que camadas totalmente conectadas. São compostas por diferentes camadas convolucionais que aprendem a reconhecer diferentes características na imagem.

**Data Augmentation:** Geração de novos dados de treinamento aplicando transformações as imagens existentes no conjunto de dados, aumentando o tamanho e junto com isso a diversidade ajudando a melhorar a precisão do modelo treinado, melhorando o desempenho na tarefa de classificação de imagens

## Seção 6: Autoencoders

**AutoEncoders:** São um tipo de rede neural que pode aprender a codificar e decodificar imagens, sendo comumente usada na visão computacional devido aos benefícios, como ser treinado sem rótulos, também são capazes de aprender a representar dados complexos em uma espaço de baixa dimensão, e por último, podem realizar a compressão de dados assim diminuindo o uso do espaço de armazenamento. Consiste em duas partes, um codificador, que recebe uma imagem de entrada e mapeia para uma representação dimensional inferior chamada espaço latente, e um decodificador que pega este espaço latente e reconstrói a imagem original.

**Variational AutoEncoders:** Incorpora a inferência bayesiana no processo de aprendizagem, gerando um vetor de média e um vetor de variância sendo usados para mostrar variáveis latentes, visando assim minimizar a reconstrução de erro, modelam a distribuição latente de forma probabilística, permitindo uma melhor generalização e a geração de dados realistas, como imagens ou textos, a partir de amostras da distribuição latente.

## Seção 7: Projetos

**Transferência de Estilo:** Esta técnica permite aplicar o estilo visual de uma imagem (como uma pintura famosa) ao conteúdo de outra imagem (como uma foto), criando obras de arte únicas que combinam o conteúdo de uma imagem com o estilo artístico de outra.

**Neural Deep Fake:** Usando redes neurais, essa técnica permite manipular vídeos e áudios para fazer uma pessoa parecer estar dizendo ou fazendo algo que, na

realidade, não aconteceu. Isso tem sido amplamente utilizado para criar vídeos falsos, conhecidos como "deep fakes", com um alto grau de realismo.

**Colorização de imagens:** Com essa técnica, conseguimos adicionar cores a fotos em preto e branco, trazendo uma nova vida a imagens antigas. A colorização automática utiliza redes neurais para prever as cores mais realistas com base no conteúdo da imagem, proporcionando uma reconstrução visual.