

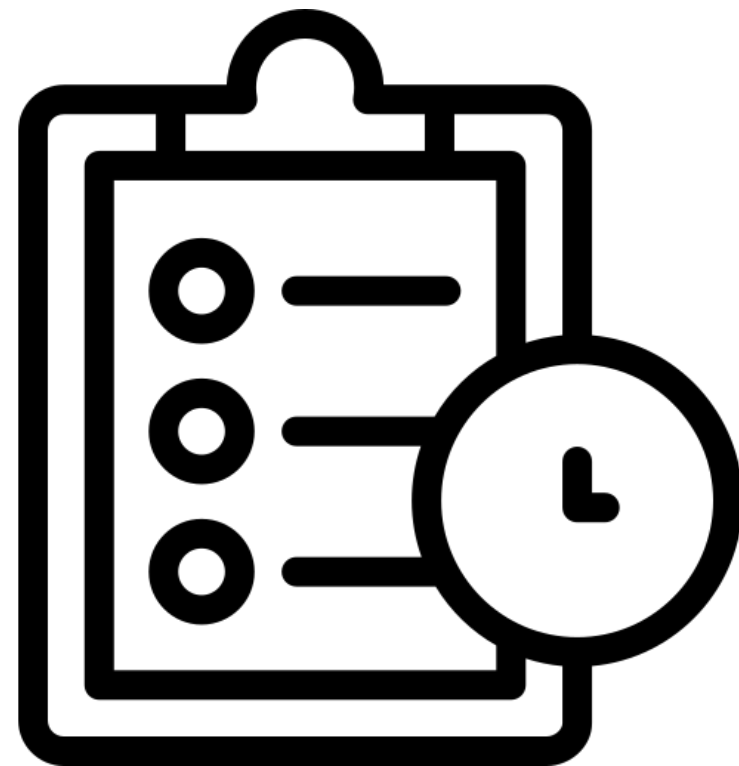


FIAP

Técnicas de Otimização e Transfer Learning

Agenda

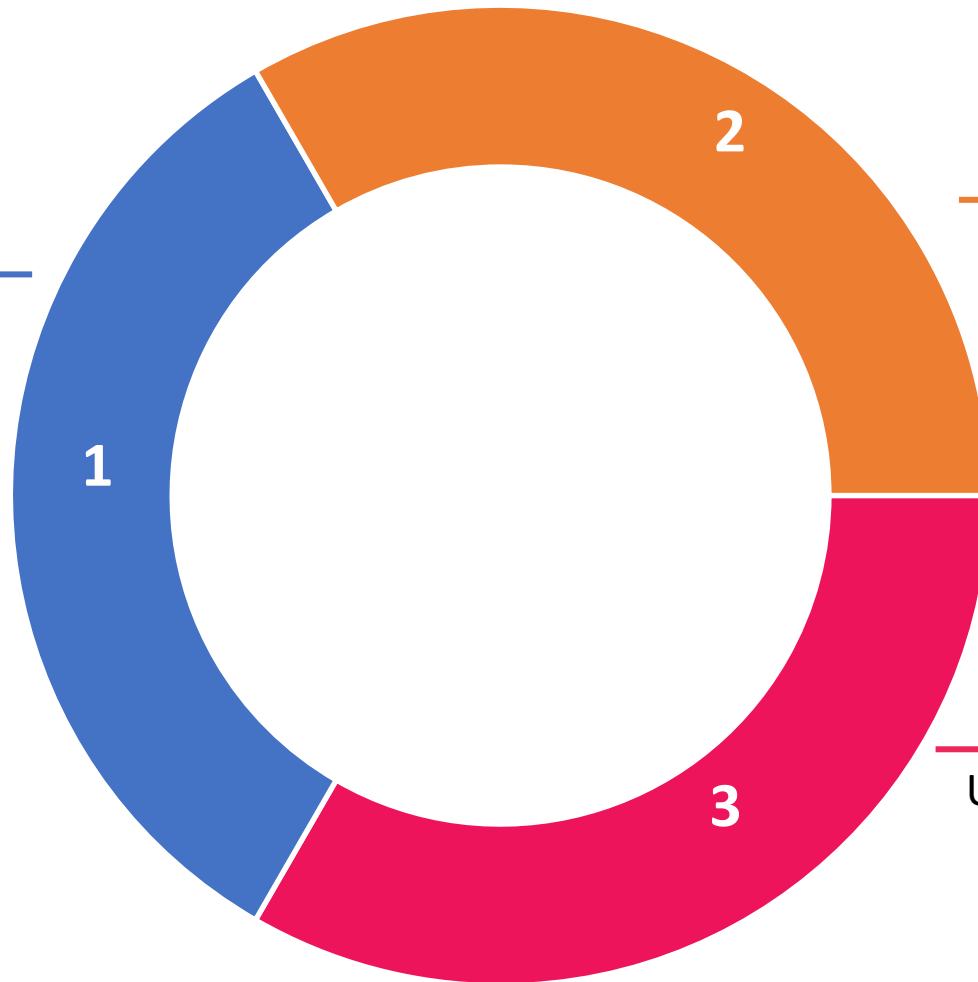
- Validation
- Dropout
- Batch Normalization
- Data Augmentation
- Laboratório de Otimização
- Arquiteturas de Redes CNN
- O que é uma Rede Pré Treinada?
- O que é o Transfer Learning?
- Combinando Rede Pré Treinada com um Classificador
- Roteiro de aplicação de Transfer Learning
- Laboratório de Transfer Learning



Validation – Conjunto de dados

1. Conjunto de Treinamento

Usado para ajustar os pesos e bias do modelo



2. Conjunto de Validação

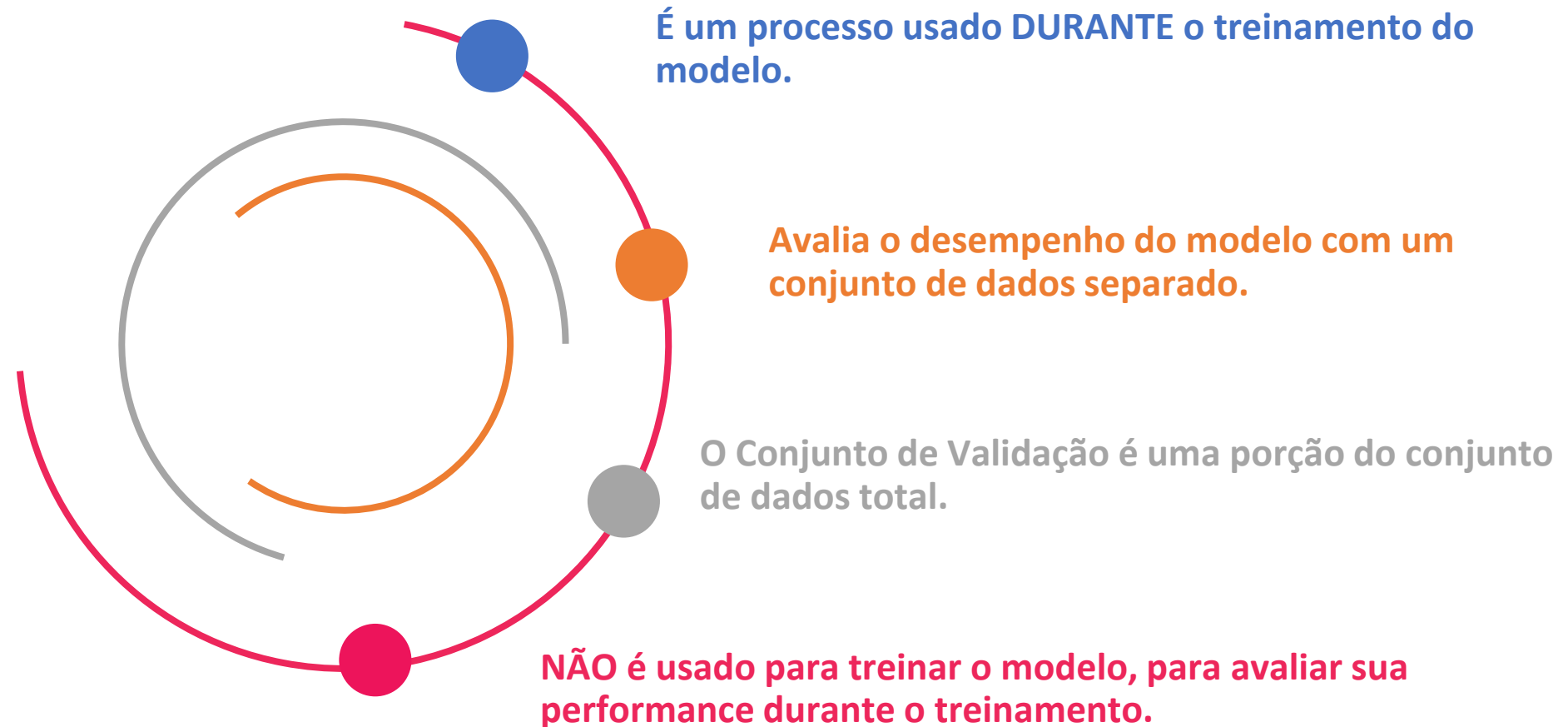
Usado para monitorar a performance do modelo **DURANTE** o treinamento

3. Conjunto de Teste

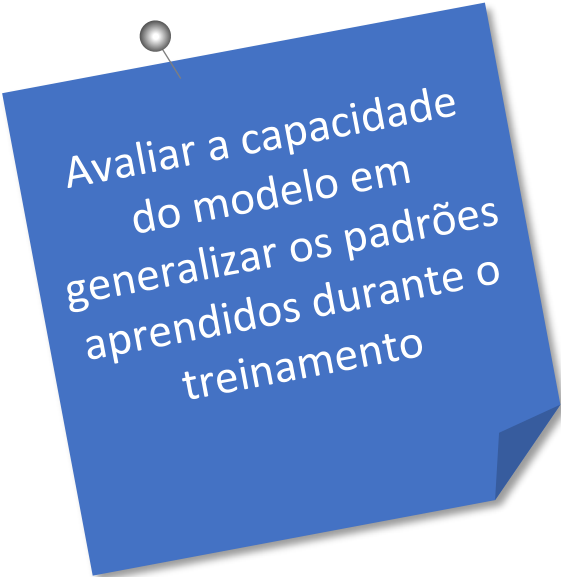
Usado para avaliar a performance do modelo **APÓS** o treinamento

Validation – Processo de Validação

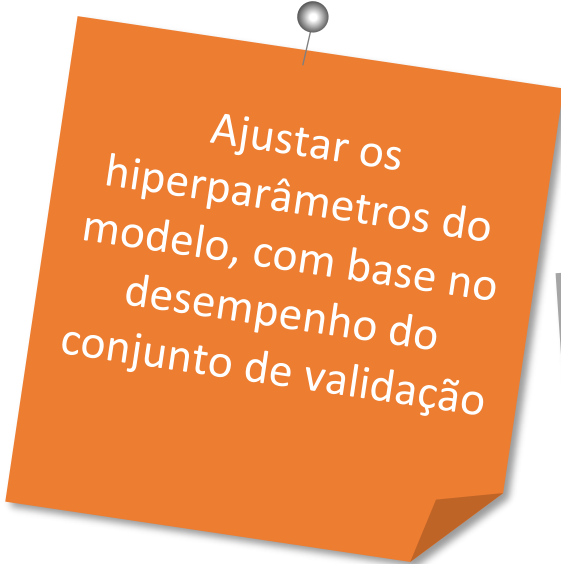
FIAP



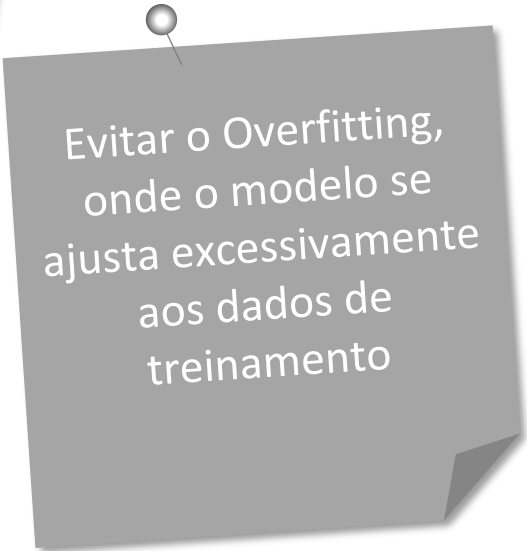
Validation – Importância



Avaliar a capacidade do modelo em generalizar os padrões aprendidos durante o treinamento



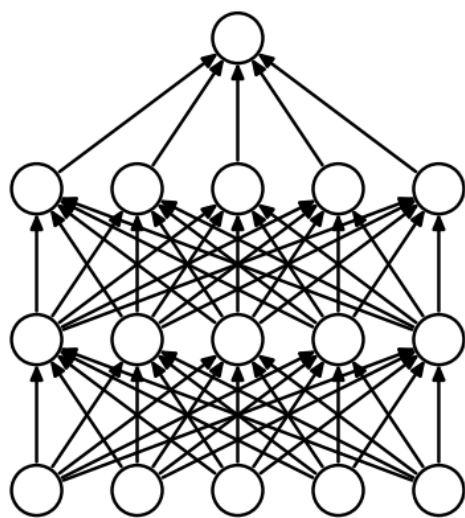
Ajustar os hiperparâmetros do modelo, com base no desempenho do conjunto de validação



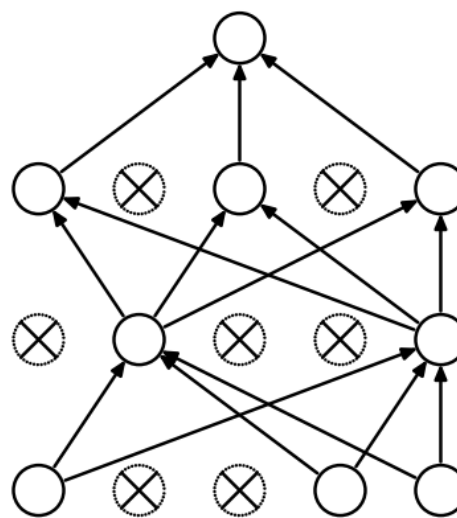
Evitar o Overfitting, onde o modelo se ajusta excessivamente aos dados de treinamento

Dropout

Consistem em, aleatoriamente, ignorar alguns neurônios da rede durante o processo de treinamento para prevenir que a Rede Neural se especialize demais nos dados de treinamento, evitando o Overfitting. Normalmente, usamos um Dropout de 20 a 30%.



(a) Standard Neural Net

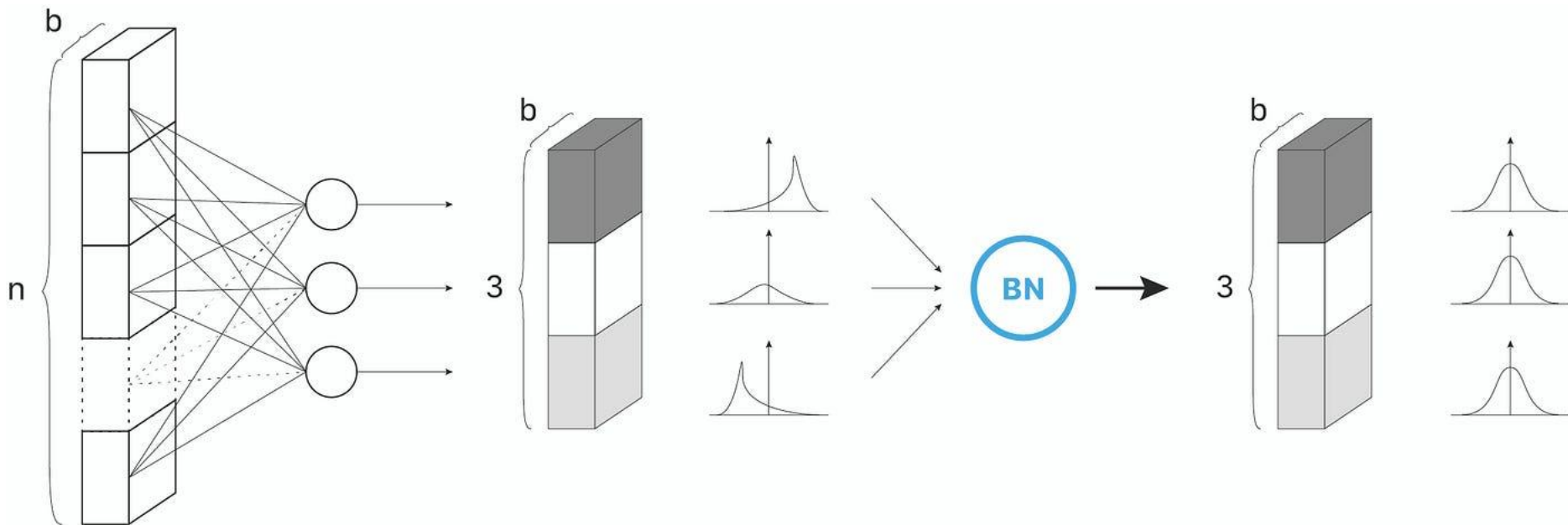


(b) After applying dropout.

Fonte: Srivastava et al. in [Dropout: A Simple Way to Prevent Neural Networks from Overfitting](#)

Nem sempre dá certo, e alguns autores até são contra o uso de Dropout.

Batch Normalization



Técnica para normalizar as ativações intermediárias durante o treinamento

Ajusta as ativações em cada camada para ter uma média zero e uma variância unitária em relação aos valores das ativações

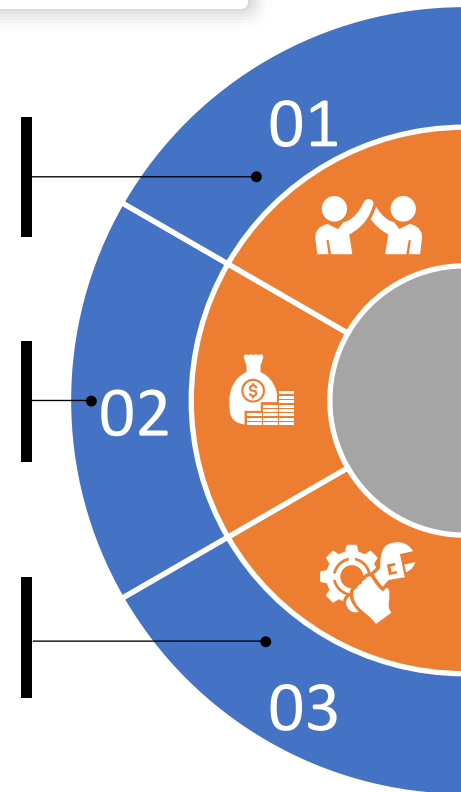
É realizada adicionando uma camada de Batch Normalization após as camadas de convolução ou após as camadas densas da rede

O Uso de Batch Normalization apresenta as seguintes vantagens:

Aceleração do treinamento: Pode ajudar a reduzir a dependência das taxas de aprendizado e inicialização dos pesos iniciais, permitindo que as redes neurais sejam treinadas com taxas de aprendizado mais altas, o que pode acelerar o treinamento.

Melhoria do desempenho: Pode ajudar a mitigar o problema da "covariate shift", que é a mudança na distribuição dos dados de entrada durante o treinamento. Isso pode melhorar a estabilidade e a generalização do modelo.

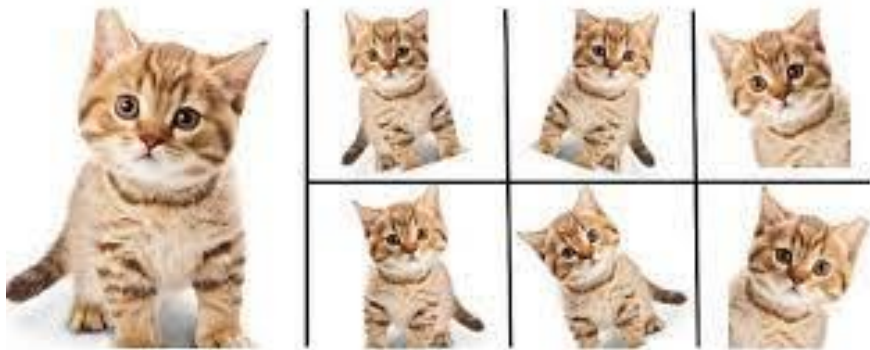
Regularização implícita: Também pode atuar como uma forma de regularização implícita, ajudando a reduzir o overfitting ao estabilizar as ativações intermediárias durante o treinamento.



Data Augmentation

Consistem em aumenta o seu dataset de imagens, gerando pequenas variações na imagem.

No exemplo abaixo, o modelo não saberá que se trata do mesmo gato e tratará como um dado diferente, mas com características semelhantes.



Enlarge your Dataset



Crop



Symmetry



Rotation



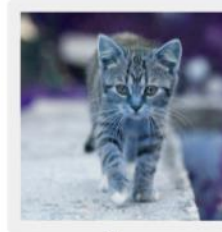
Scale



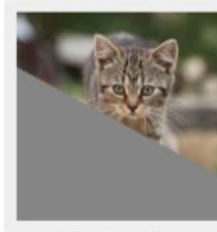
Original



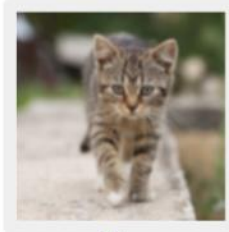
Noise



Hue



Obstruction

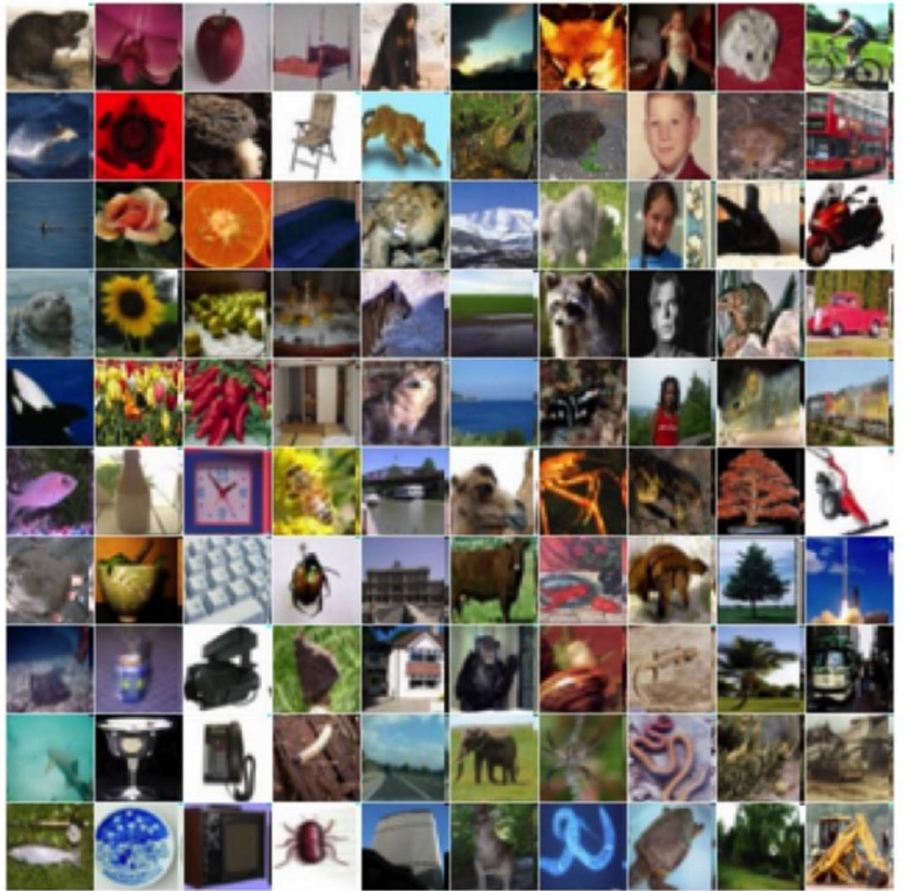


Blur

Laboratório

O CIFAR-100 é um conjunto de dados de referência amplamente utilizado em problemas de visão computacional e aprendizado de máquina. Ele contém 60.000 imagens coloridas de 32x32 pixels, divididas em 100 classes distintas. Cada classe possui 600 imagens, sendo 500 para treinamento e 100 para teste. O CIFAR-100 é um desafio mais complexo em comparação com o CIFAR-10, pois envolve classificação em um espaço de classe mais amplo. É usado como um benchmark para avaliar a capacidade de modelos de aprendizado de máquina em lidar com uma ampla variedade de classes e objetos do mundo real.

O objetivo deste laboratório é tentar **melhorar** um modelo de Rede Neural Convolucional para identificar as classes das imagens, usando as **técnicas de otimização**.



Arquiteturas de Redes CNN



Existem diversas arquitetura de CNN, cada uma com suas próprias características, principalmente para visão computacional. Mas todas terão em comum camadas de convolução e maxpooling, além das técnicas de otimização que vimos anteriormente.



E por que usar uma arquitetura de Rede CNN já existente?

1

Utilizar uma arquitetura de CNN possibilita reduzir o tempo de pesquisa com o desenvolvimento de novas arquiteturas uma vez que essas arquiteturas já foram sistematicamente revisadas.

2

O treinamento de uma boa CNN não é simples, além de muitos dados (milhares de imagens) e muito tempo de processamento

3

Uma forma de contornar esse problema é a utilização de redes pré-treinadas com conjunto de dados de milhares de imagens, o que garante uma boa acurácia.

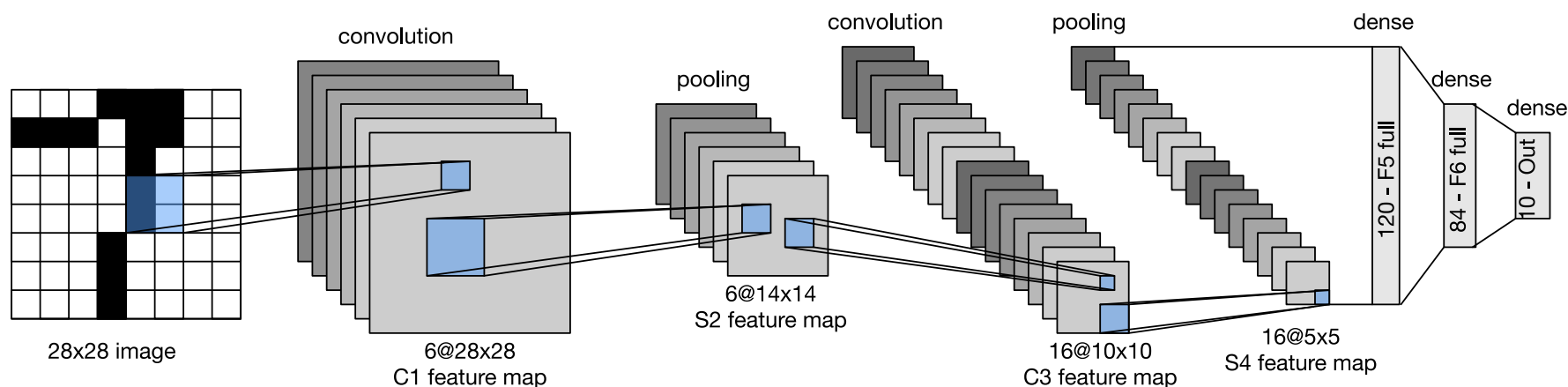
4

É possível ajustar os pesos das últimas camadas da rede para detectar apenas os recursos relevantes para o problema específico

Arquiteturas de Redes CNN



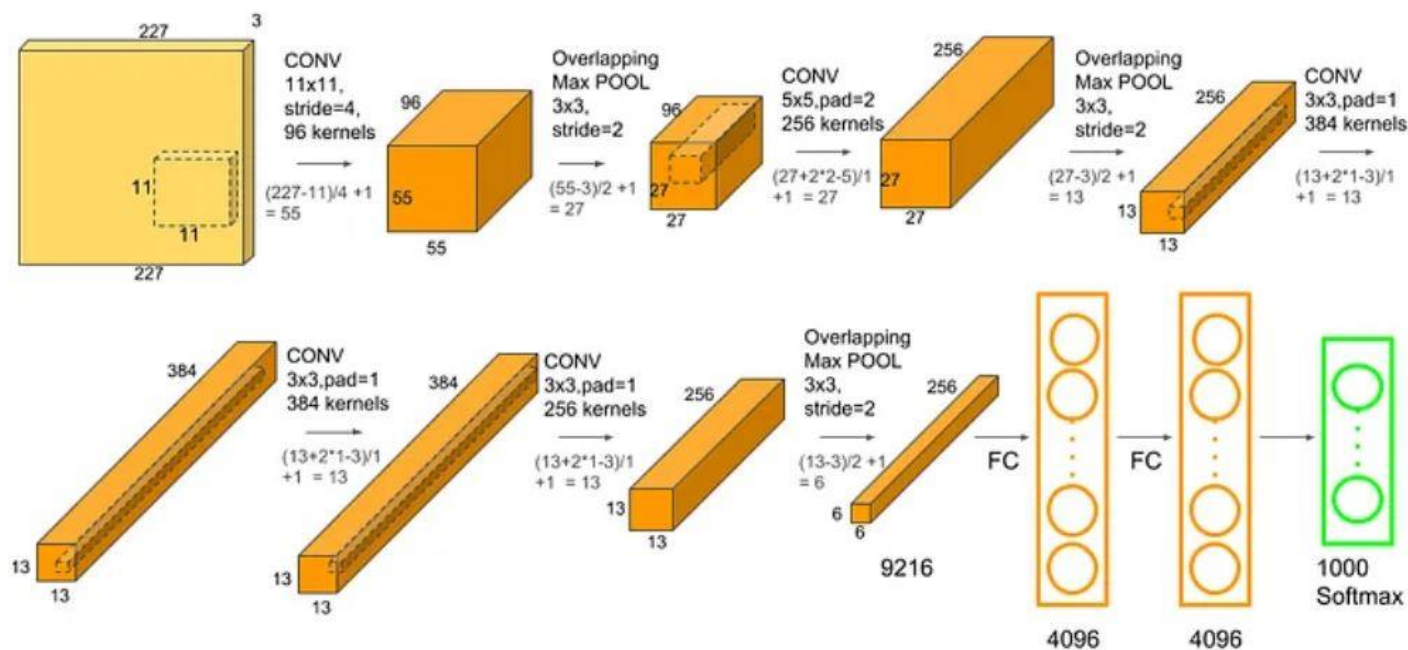
LeNET: Desenvolvida em 1998 por Yann LeCun, a LeNet foi pioneira no uso de camadas de convolução com filtros 5x5 e passo 1, além de camadas de agrupamento com filtros 2x2 e passo 2, intercaladas por camadas totalmente conectadas (FC). A ordem das camadas é: CONV-POOL-CONV-POOL-FC-FC. Essa arquitetura teve um papel fundamental no reconhecimento de dígitos manuscritos.



Arquiteturas de Redes CNN



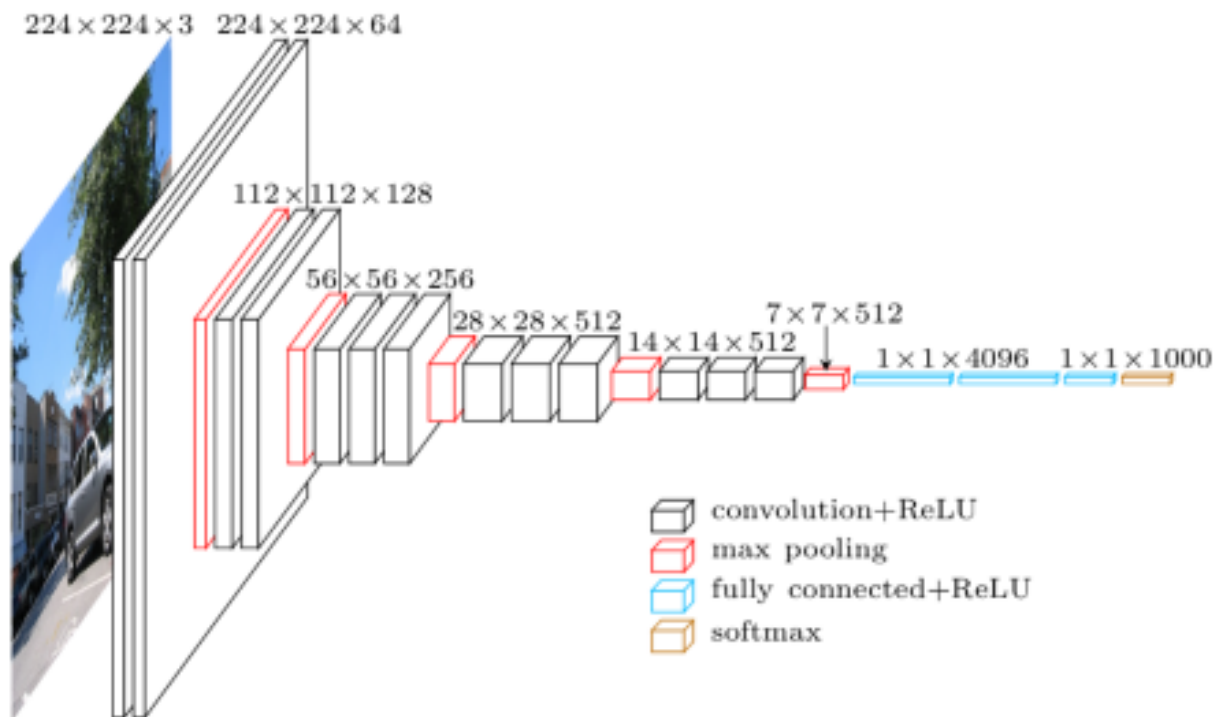
AlexNet: Criada em 2012 por Alex Krizhevsky, Ilya Sutskever e Geoffrey Hinton, a AlexNet é uma arquitetura mais avançada que a LeNet. Possui cinco camadas convolucionais seguidas de três camadas FC, e emprega a função de ativação ReLU. Vencedora da competição ImageNet de 2012, marcou o início da popularização das redes neurais convolucionais profundas.



Arquiteturas de Redes CNN



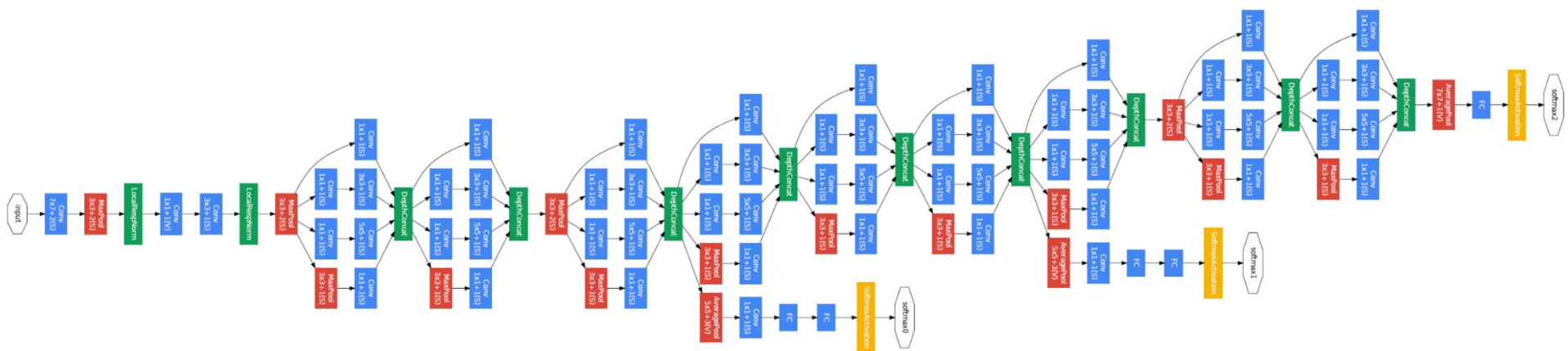
VGG: Concebida em 2014 pelo Visual Geometry Group da Universidade de Oxford, propôs o uso de filtros menores (3x3) em redes mais profundas, com no mínimo 16 camadas convolucionais e maxpooling com filtros 2x2. Apesar de os filtros menores gerarem menos parâmetros, as camadas FC e as convoluções iniciais demandavam grande quantidade de memória RAM, resultando em uma rede pesada



Arquiteturas de Redes CNN



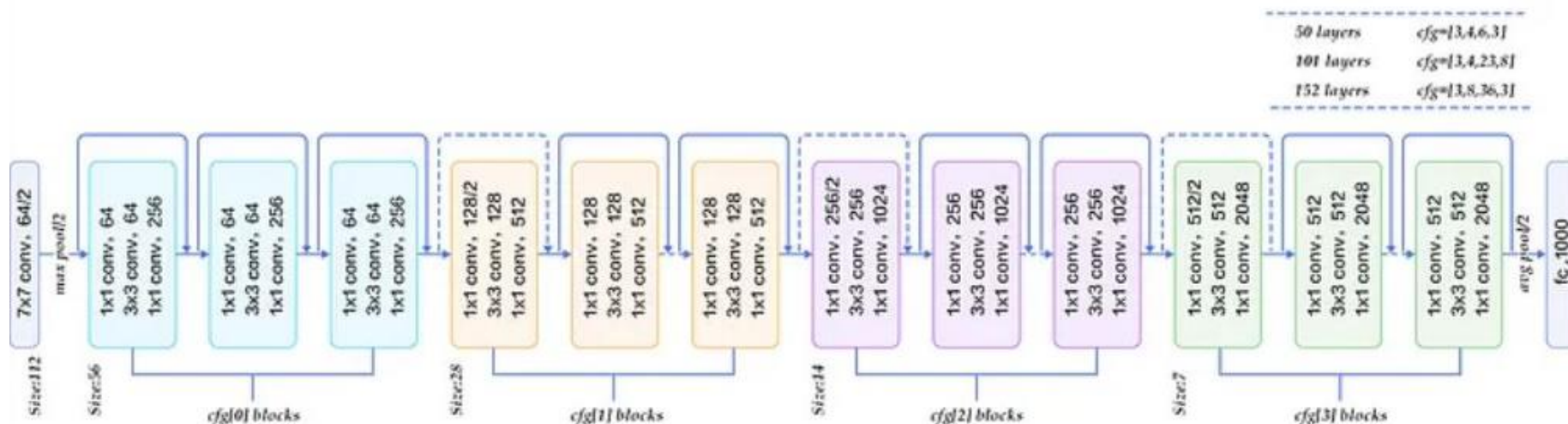
GoogleNET: Paralelamente à VGG, em 2014, pesquisadores do Google desenvolveram a GoogleNet, que introduziu o módulo Inception como elemento fundamental. Com nove módulos Inception em sequência, a arquitetura utiliza convoluções 3x3 e 5x5 precedidas por convoluções 1x1 para diminuir o custo computacional. A GoogleNet foi projetada para ser eficiente em termos de recursos e venceu a competição ImageNet de 2014.



Arquiteturas de Redes CNN



ResNET: A rede residual, proposta em 2015 por Kaiming He e colaboradores, tem como característica principal a inclusão de conexões residuais (curto-circuitos) a cada duas convoluções, adicionando um resultado anterior ao resultado futuro. Isso permite treinar redes mais profundas sem problemas de degradação do desempenho. ResNets com 50, 101 e 152 camadas utilizam blocos residuais com "bottleneck", que consistem em duas convoluções 3x3 intercaladas por convoluções 1x1, diminuindo o custo computacional.





MobileNET: Proposta em 2017, é uma arquitetura otimizada para dispositivos móveis e aplicativos com limitações de recursos computacionais. Utiliza convoluções separáveis por profundidade para reduzir o número de parâmetros e o consumo de memória.



EfficientNET: Proposta em 2019, é uma família de redes neurais convolucionais que busca melhorar a eficiência em termos de recursos computacionais e desempenho, através do ajuste coordenado da largura, profundidade e resolução das redes.



InceptionV3: Uma evolução do GoogleNet, a InceptionV3 é uma arquitetura desenvolvida em 2015 que aprimora o módulo Inception e implementa técnicas de normalização em lotes. Essa arquitetura alcança um desempenho superior com menos parâmetros e menor custo computacional.



DenseNet: Proposta em 2016, a DenseNet é uma arquitetura que introduz conexões densas entre as camadas. Cada camada recebe as características de todas as camadas anteriores, o que melhora o fluxo de informações e gradientes durante o treinamento. Isso permite a construção de redes mais profundas e eficientes.



YOLO: (You Only Look Once): É uma arquitetura de rede neural focada em detecção de objetos em tempo real. Proposta em 2016, a YOLO divide a imagem em regiões e prevê, de uma só vez, as probabilidades de classes e as coordenadas das caixas delimitadoras. A YOLO é conhecida por sua velocidade e capacidade de detectar objetos em tempo real.



Transforme: Embora não seja uma arquitetura de rede neural convolucional, o Transformer, proposto em 2017, é uma arquitetura de rede neural notável para processamento de linguagem natural e outras tarefas sequenciais. O Transformer introduziu o conceito de atenção auto-regressiva, que permite que a rede aprenda relacionamentos complexos entre as entradas, e tem sido a base para modelos de linguagem de última geração, como BERT e GPT.

O que é Transfer Learning

Transfer Learning (Aprendizagem por Transferência) é uma abordagem que nos permite tirar proveito das arquiteturas de redes neurais existentes e treiná-las para classificar objetos personalizados ou novas categorias de imagens.

É uma técnica em que um modelo de aprendizado profundo, treinado previamente em um conjunto de dados maior e mais diversificado, é adaptado para ser aplicado a um novo problema. O conhecimento adquirido pelo modelo original é transferido para o novo problema, permitindo um treinamento mais rápido e, muitas vezes, um desempenho melhor do que treinar uma rede neural do zero.

A ideia é que as redes pré-treinadas, já aprenderam a extrair características importantes das imagens em seus primeiros estágios. Essas características podem ser comuns a muitos problemas de classificação de imagens, como detecção de bordas, texturas e padrões. Ao aproveitar esse conhecimento, podemos nos concentrar no treinamento das últimas camadas do modelo, que são responsáveis por aprender características específicas do novo problema.

Ao utilizar o Transfer Learning, podemos economizar tempo e recursos computacionais, além de obter melhores resultados do que treinar uma rede do zero para um conjunto de dados menor e específico. Portanto, é uma técnica amplamente utilizada em aplicações práticas de aprendizado profundo e processamento de imagens.

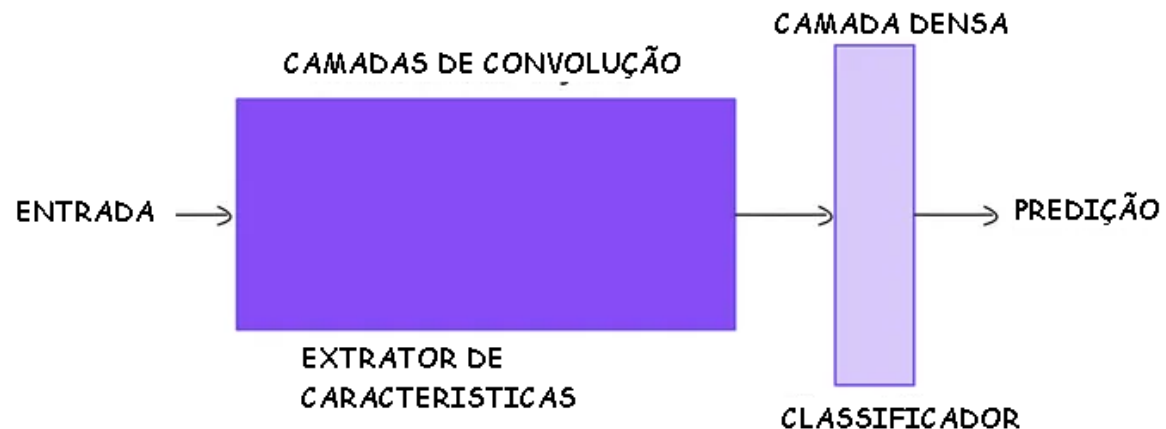
Combinando a Rede Pré Treinada com um Classificador

Ao aplicar o Transfer Learning, nossa rede convolucional será composta por duas partes principais: o extrator de características e o classificador.

O extrator de características será baseado em uma rede pré-treinada, como VGG16, ResNet50 ou InceptionV3. Essa parte da rede já aprendeu a extrair características relevantes de imagens, como bordas, texturas e padrões, durante o treinamento em um grande conjunto de dados, como o ImageNet.

Em seguida, adicionaremos um classificador MLP (Multilayer Perceptron) personalizado para resolver o nosso problema específico de classificação de imagens. Esse classificador será responsável por aprender as características específicas do novo conjunto de dados e classificar as imagens nas categorias desejadas.

Dessa forma, a rede ajustada combina o poder das redes pré-treinadas, que já aprenderam a extrair características gerais de imagens, com um classificador personalizado que aprenderá a distinguir as categorias específicas do nosso problema. Como mostra a figura abaixo:



Roteiro de Aplicação de Transfer Learning

1 Escolha uma rede pré treinada

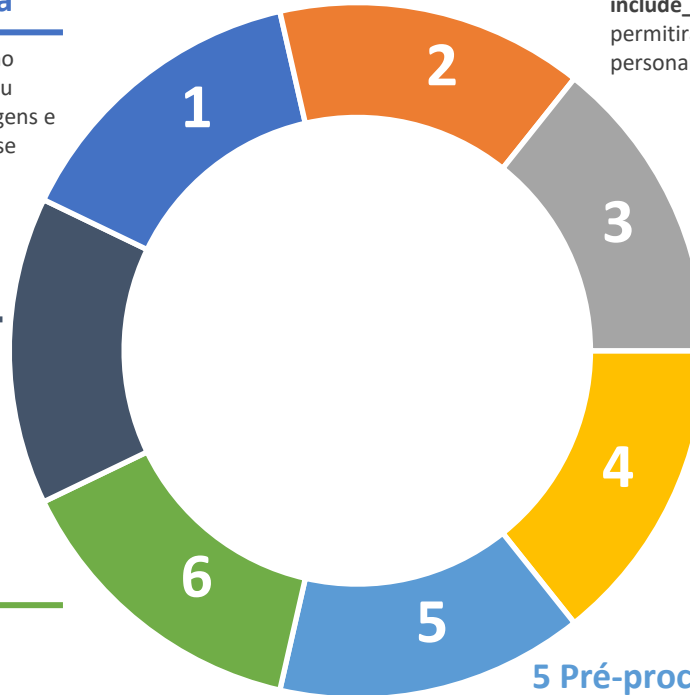
Selecione uma rede neural pré-treinada, disponível no Keras, com base nas características e requisitos do seu problema. Cada arquitetura tem suas próprias vantagens e desvantagens, portanto, escolha aquela que melhor se adapta às suas necessidades.

7 Avalie e otimize:

Avalie o desempenho do modelo ajustado em um conjunto de teste e otimize os hiperparâmetros conforme necessário. Você pode experimentar diferentes arquiteturas de redes neurais, taxas de aprendizado, otimizadores e outros hiperparâmetros para encontrar a melhor configuração para o seu problema.

6 Treine o modelo:

Treine o modelo ajustado no seu conjunto de dados. Durante as primeiras épocas, com as camadas pré-treinadas congeladas, o modelo aprenderá as características específicas do novo problema.



2 Remova a camada de classificação

Carregue a rede neural pré-treinada sem a camada de classificação final. Isso pode ser feito usando o argumento **include_top = False** ao carregar o modelo no Keras. Isso permitirá que você adicione suas próprias camadas personalizadas para classificar as novas categorias.

3 Adicione camadas personalizadas:

Adicione camadas específicas para o seu problema de classificação. Normalmente, isso inclui uma camada de **GlobalAveragePooling2D**, seguida por uma **camada densa** com uma função de ativação **softmax** e o número de neurônios igual ao número de classes do novo problema.

4 Congele as camadas pré-treinadas

É uma boa prática congelar as camadas pré-treinadas da rede neural, especialmente durante as primeiras épocas do treinamento. Isso evitará que os pesos dessas camadas sejam atualizados e preservará o conhecimento prévio que elas possuem. No Keras, você pode fazer isso com o **model.trainable = False**

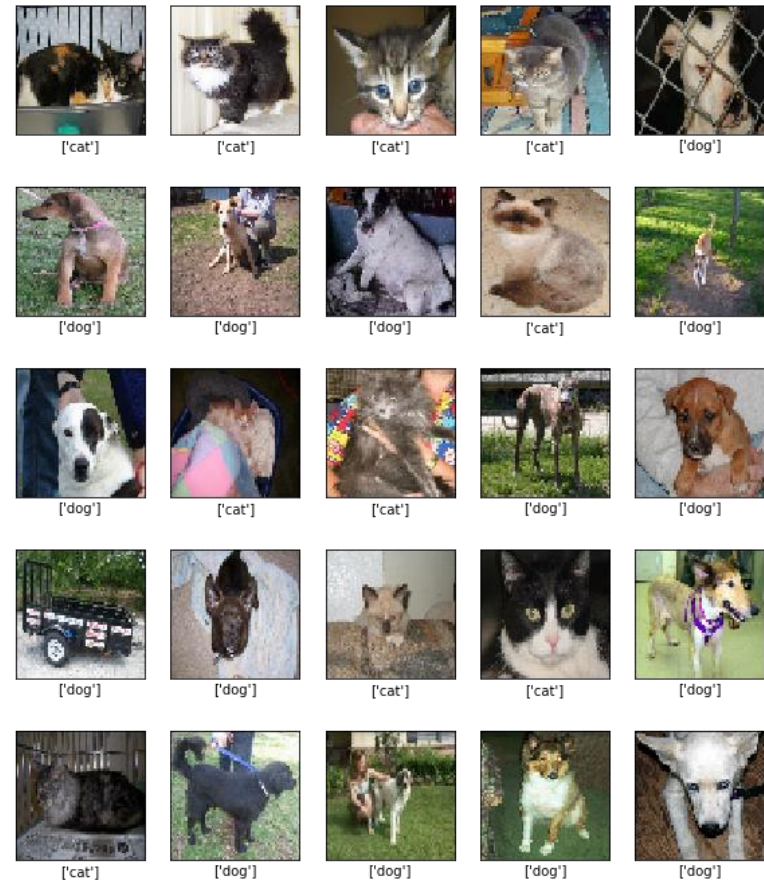
5 Pré-processamento dos dados:

Prepare os dados de acordo com a rede pré-treinada escolhida. Isso inclui redimensionar as imagens, normalizar os valores dos pixels e codificar as etiquetas das categorias. **Lembre-se de aplicar as mesmas transformações usadas no conjunto de dados original da rede pré-treinada.**

Laboratório de Transfer Learning

O dataset "cats_vs_dogs" do TensorFlow é um conjunto de dados amplamente utilizado para tarefas de classificação de imagens binárias de gatos e cachorros. Consiste em milhares de imagens coloridas de gatos e cachorros, onde cada imagem é associada a um rótulo indicando se é um gato ou um cachorro.

O objetivo deste laboratório é aplicar a técnica de **Transfer Learning** para otimizar o processo de classificação das imagens deste dataset.



Copyright © 2023 Prof. Airtton Y. C. Toyofuku/Yan Coelho

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).