



Implementação e Análise de Classificação com Redes Convolucionais e o dataset CUFS

Alunas: Clara Raquel Santos e Jemyma Kesly Silva

Data de entrega: 03/12/2024

Jequié -BA

2024

Resumo

Implementação e Análise de classificação com Redes Convolucionais e o dataset CUFS

Objetivo da atividade:

O principal objetivo desta atividade foi explorar o funcionamento das Redes Convolucionais, usando para isso o *dataset* CUHK Face Sketch Database (CUFS).

Aqui foram propostos três diferentes modelos CNNs para o conjunto desses dados, todos com diferentes combinações de tamanho e número de filtros, funções de ativação, técnicas de normalização, funções para minimizar a perda, etc. Todos os modelos utilizaram o otimizador 'Adam' já que este algoritmo é bem popular por combinar as melhores características de outros otimizadores (como o RMSprop) e converge mais rápido que outros como ele.

Os resultados foram interessantes já que o segundo modelo demonstrou como é fácil para uma Rede Neural Convolucional se aderir aos dados usados de forma exagerada, causando *Overfitting* e o terceiro provou como o custo computacional de um modelo pode até mesmo inviabilizar sua utilização de forma cotidiana.

De forma geral, todos os modelos se saíram bem, com acurácia acima de 70% para a maioria dos casos e uma relativa baixa perda. Porém, por conta dos problemas supracitados, o melhor desempenho foi atribuído ao primeiro modelo proposto.

Contextualizando:

Objetivo das CNNs

A implementação e análise de classificação com redes convolucionais (CNNs) visa desenvolver e avaliar modelos de aprendizado profundo capazes de categorizar dados, principalmente imagens. O objetivo central é construir um sistema que, dado um conjunto de dados de entrada, possa atribuir cada elemento a uma classe pré-definida com alta precisão.

Como deverão ser realizadas as avaliações:

A avaliação de um modelo de CNN envolve as seguintes etapas:

1. Preparação dos dados:

- Coleta: Reunião de um conjunto de dados representativo e balanceado, com imagens de alta qualidade e rotuladas corretamente.
- Pré-processamento: Aplicação de técnicas para transformar as imagens em um formato adequado para a rede neural, como redimensionamento, normalização e conversão para um tensor numérico.
- Divisão: Separação dos dados em três conjuntos: treinamento, validação e teste. O conjunto de treinamento é utilizado para ajustar os pesos da rede, o de validação para monitorar o desempenho durante o treinamento e o de teste para avaliar o desempenho final do modelo em dados nunca vistos antes.

2. Construção da arquitetura da rede:

- Escolha da arquitetura: Seleção de uma arquitetura de CNN adequada à tarefa e ao tamanho do conjunto de dados. Existem diversas arquiteturas pré-treinadas disponíveis, como VGG, ResNet e Inception, ou pode-se criar uma arquitetura personalizada.
- Definição das camadas: A rede é composta por camadas convolucionais, de pooling, de ativação e totalmente conectadas. Cada camada desempenha um papel específico na extração de características e na classificação.

3. Treinamento:

- Escolha da função de perda: Definição de uma função de perda que mede a diferença entre as previsões do modelo e os rótulos verdadeiros.
- Escolha do otimizador: Seleção de um algoritmo de otimização para ajustar os pesos da rede, como o Stochastic Gradient Descent (SGD) ou o Adam.
- Treinamento: Apresentação repetida dos dados de treinamento à rede, ajustando os pesos a cada iteração para minimizar a função de perda.

4. Avaliação:

- Métricas de desempenho: Cálculo de métricas como acurácia, precisão, revocação e F1-score para avaliar o desempenho do modelo no conjunto de teste.
- Análise de erros: Identificação dos tipos de erros cometidos pelo modelo para entender suas limitações e propor melhorias.

Principais resultados:

Os principais resultados esperados de uma implementação de classificação com CNNs incluem:

- Alta acurácia: O modelo deve ser capaz de classificar corretamente uma alta porcentagem das imagens do conjunto de teste.
- Generalização: O modelo deve ser capaz de generalizar para dados nunca vistos antes, evitando o overfitting.
- Interpretabilidade: Em alguns casos, é desejável que o modelo seja interpretável, ou seja, que seja possível entender quais características das imagens o modelo está utilizando para realizar a classificação.
- Eficiência: O modelo deve ser capaz de realizar a classificação em tempo real ou próximo ao tempo real, dependendo da aplicação.

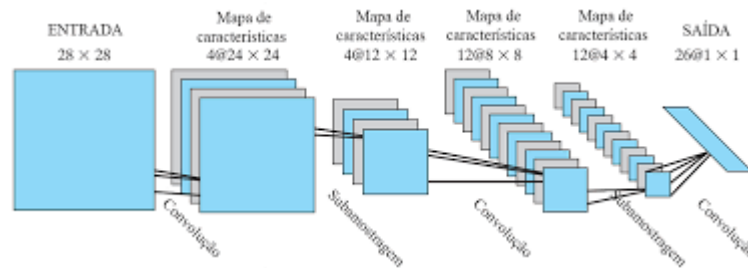
Aplicações:

As CNNs são amplamente utilizadas em diversas áreas, como:

- Visão computacional: Reconhecimento facial, detecção de objetos, segmentação de imagens.

- Processamento de linguagem natural: Análise de sentimentos, classificação de textos.
- Medicina: Análise de imagens médicas, detecção de doenças.
- Automação industrial: Inspeção de qualidade, controle de robôs.

De forma prática e eficiente , a implementação e análise de classificação com CNNs é um processo iterativo que envolve a construção, treinamento e avaliação de modelos de aprendizado profundo para resolver problemas de classificação de dados, principalmente imagens. A escolha da arquitetura, dos hiperparâmetros e das técnicas de avaliação são cruciais para o sucesso do modelo



Introdução

Redes Neurais Convolucionais

Uma Rede Neural Convolucional (ConvNet / Convolutional Neural Network / CNN) é um algoritmo de Aprendizado Profundo que pode captar uma imagem de entrada, atribuir importância (pesos e vieses que podem ser aprendidos) a vários aspectos / objetos da imagem e ser capaz de diferenciar um do outro. O pré-processamento exigido em uma ConvNet é muito menor em comparação com outros algoritmos de classificação. Enquanto nos métodos primitivos os filtros são feitos à mão, com treinamento suficiente, as ConvNets têm a capacidade de aprender esses filtros / características.

As Redes neurais convolucionais são uma variante das redes neurais artificiais, as quais se mostraram eficientes no reconhecimento e classificação de imagens, sendo uma ferramenta útil, em especial quando aliadas à visão computacional, para o desenvolvimento de aplicações embutidas em robôs e carros autônomos. Segundo Geitgey (2016), estas são capazes de reduzir a complexidade no tratamento de imagens utilizando camadas de convolução e filtros, as quais simplificam a imagem através de operações de convolução e agrupamento.

A arquitetura de uma ConvNet é análoga àquela do padrão de conectividade de neurônios no cérebro humano e foi inspirada na organização do Visual Cortex. Os neurônios individuais respondem a estímulos apenas em uma região restrita do campo visual conhecida como Campo Receptivo. Uma coleção desses campos se sobrepõem para cobrir toda a área visual.

As Redes Convolucionais são um tipo específico de rede neural profunda, projetadas para trabalhar com dados que possuem uma estrutura de grade, como imagens. Elas são inspiradas no funcionamento do córtex visual humano, onde células nervosas respondem a estímulos específicos em um campo receptivo.

Por que usar CNNs para classificação de imagens

- **Extração automática de características:** As CNNs aprendem a extrair as características mais relevantes de uma imagem diretamente dos dados, sem a necessidade de engenharia de features manual.
- **Invariância a translações:** As CNNs são capazes de reconhecer objetos em diferentes posições na imagem.
- **Hierarquia de características:** As camadas convolucionais aprendem a extrair características de baixo nível (como bordas) nas camadas iniciais e características de alto nível (como objetos completos) nas camadas mais profundas.

Processo de Classificação com CNNs

1. **Pré-processamento:** As imagens são convertidas em um formato numérico (e.g., matrizes NumPy) e normalizadas para um intervalo específico.
2. **Convolução:** Filtros são aplicados à imagem para extrair características. A saída de cada filtro é um mapa de características.
3. **Pooling:** Reduz a dimensionalidade dos mapas de características, tornando o modelo mais eficiente e invariante a pequenas translações.
4. **Flatten:** A saída das camadas convolucionais e de pooling é achatada em um único vetor.
5. **Classificação:** As camadas totalmente conectadas recebem o vetor achatado e produzem uma probabilidade para cada classe.

Implementação

Ferramentas:

- **TensorFlow:** A biblioteca de aprendizado de máquina mais popular, oferecendo flexibilidade e recursos avançados.
- **Keras:** Uma API de alto nível construída sobre o TensorFlow, facilitando a criação e treinamento de modelos.
- **PyTorch:** Uma alternativa ao TensorFlow, com um estilo de programação mais imperativo e uma comunidade ativa.

Etapas:

1. **Carregar os dados:** Carregar as imagens e seus rótulos correspondentes.
2. **Criar o modelo:** Definir a arquitetura da CNN, incluindo o número de camadas convolucionais, o tamanho dos filtros, o tipo de pooling e o número de neurônios nas camadas totalmente conectadas.
3. **Compilar o modelo:** Especificar a função de perda, o otimizador e as métricas de avaliação.
4. **Treinar o modelo:** Apresentar as imagens ao modelo e ajustar os pesos para minimizar a função de perda.
5. **Avaliar o modelo:** Avaliar o desempenho do modelo em um conjunto de dados de teste usando métricas como acurácia, precisão, recall e F1-score.

Análise dos Resultados

- **Matriz de confusão:** Visualizar quais classes são mais facilmente confundidas.
- **Curvas ROC:** Avaliar o desempenho do modelo em diferentes limiares de classificação.
- **Visualização de filtros:** Entender quais características as camadas convolucionais estão aprendendo.
- **Análise de erros:** Identificar exemplos em que o modelo comete erros e tentar entender as razões.

A Importância das Redes Convolucionais

As redes convolucionais (CNNs, do inglês *Convolutional Neural Networks*) revolucionaram o campo da visão computacional e se tornaram uma ferramenta essencial em diversas áreas da inteligência artificial. Sua importância reside em sua capacidade de extrair automaticamente características relevantes de imagens e classificá-las com alta precisão.

Por que as CNNs são tão importantes nesse processo

- **Extração de características:** Ao contrário de métodos tradicionais, que dependiam de engenharia de características manuais, as CNNs aprendem automaticamente as características mais relevantes de uma imagem, como bordas, texturas e padrões.
- **Invariância a translações:** As CNNs são capazes de identificar objetos em diferentes posições dentro da imagem, tornando-as robustas a pequenas variações na localização dos objetos.
- **Hierarquia de características:** As CNNs constroem representações hierárquicas, começando por características de baixo nível (bordas, texturas) e progredindo para características de alto nível (objetos completos), o que permite a classificação de imagens complexas.
- **Aplicações diversificadas:** As CNNs são utilizadas em diversas áreas, como:
 - **Reconhecimento de imagens:** Identificação de objetos, rostos, cenas, etc.
 - **Deteção de objetos:** Localização e classificação de objetos em imagens.
 - **Segmentação de imagens:** Divisão de uma imagem em regiões semânticas.
 - **Geração de imagens:** Criação de novas imagens a partir de exemplos.
 - **Análise médica:** Deteção de doenças em imagens médicas.

- **Veículos autônomos:** Detecção de pedestres, sinais de trânsito, etc.

Como as CNNs funcionam

As CNNs são compostas por camadas de neurônios que processam a imagem de forma hierárquica. As principais camadas são:

- **Camadas convolucionais:** Aplicam filtros (kernels) sobre a imagem, extraíndo características locais.
- **Camadas de pooling:** Reduzem a dimensionalidade dos mapas de características, tornando o modelo mais eficiente e invariante a pequenas translações.
- **Camadas totalmente conectadas:** Classificam as representações de alto nível geradas pelas camadas anteriores

Metodologia:

Como foi sugerido, primeiramente as imagens foram redimensionadas (250x200) e renomeadas (0 para 'Homem' e 1 para 'Mulher'), seguindo o modelo de classificação proposto. Os valores RGB foram divididos por 255 para que assim ficassem todos no intervalo $[0,1]$, facilitando os cálculos da máquina.

Logo após, o conjunto de dados foi separado em teste, treino e validação, dessa forma assegurando uma acurácia na avaliação dos modelos.

No primeiro modelo foi utilizado uma rede convolucional de três camadas adicionada a uma função de ativação Sigmóide, já que a classificação em questão é binária.

Já no segundo modelo, embora o número de camadas tenha sido o mesmo, a função de ativação foi modificada para a função Softmax e o número de filtros passou de 32 do modelo 1 para 9 e 18 no modelo 2. Com isso, esperava-se avaliar se o gasto computacional poderia ser reduzido.

No terceiro e último modelo, foram acrescentadas mais camadas, o que aumentou muito o custo computacional do projeto, fazendo com que o computador gastasse toda sua memória RAM e não conseguisse executar o segundo modelo Convolucional até que duas camadas fossem retiradas e que o valor da camada densa fosse também reduzido.

Discussão:

Há 133 imagens de homens e apenas 55 imagens de mulheres neste *database*. É provavelmente por isso que dois dos três modelos (o segundo e o terceiro) se ajustaram fortemente e apresentam a tendência de classificar a maior parte das imagens como sendo de homens, mesmo quando elas não são.

Os resultados das matrizes de confusão demonstram que os modelos são melhores em classificar corretamente uma imagem que não é de um homem (ou de uma mulher) do que imagens de são de homens ou de mulheres. Uma hipótese para essa causa é o número baixo de amostras (somente 188 imagens no total).

Como um todo, os três modelos obtiveram boa acurácia (todos na faixa dos 70%), porém, provavelmente devido ao baixo número de imagens (mais uma vez) o número de perda também foi elevado (cerca de 40% para o primeiro modelo e 60% para o último).

Devido ao maior número de camadas do terceiro modelo, acreditava-se que este seria o melhor. Foi uma surpresa perceber que este apresentava um alto custo computacional, chegando ao ponto de inviabilizar os cálculos no Google Colab, que simplesmente se desconectava quando a memória RAM do computador era consumida completamente.

Após diversas modificações, entre elas a subtração de algumas camadas e a diminuição do valor da camada de densidade, o modelo finalmente conseguiu ser concluído. Mesmo assim, ele ainda apresenta falhas em sua classificação.

Conclusão e Trabalho futuros

Com essa experiência foi possível avaliar os conhecimentos adquiridos através dos meses e ver na prática a importância das Redes Convolucionais além de seu poder de classificação.

Com o aprendizado de *deep learning* foi possível reconhecer como a tecnologia evoluiu e evolui todos os dias, servindo a humanidade das mais diversas formas e auxiliando-a a minimizar o gasto com tempo e com os erros que são tão comumente realizados por pessoas ao redor do mundo.

Esse desafio serviu o propósito de abrir caminho para uma área de extrema importância da computação, e também demonstrou como as CNNs podem ser usadas cotidianamente para a resolução de problemas pelos quais a maioria não está ciente e até mesmo para satisfazer curiosidades de qualquer um que esteja disposto a estudar.

Como sugestão para o problema apresentado, é sugerido que seja utilizado um banco maior de dados, adicionando também pessoas de outras etnias e com faixas etárias distintas e mais abrangentes. Dessa forma, será possível testar a hipótese de que características físicas similares contribuíram para que os modelos não conseguissem diferenciar tão bem os rostos femininos dos masculinos neste caso.

Referências:

▶ Build a Deep CNN Image Classifier with ANY Images

▶ Image Classification using CNN Keras | Full implementation

▶ Classificação de Placas com TensorFlow e Keras CNN | Deep Learning

▶ How to load and preprocess images from a dataset using Colab, Python, and T...

https://formiga.ifmg.edu.br/documents/2019/Biblioteca/TCCs/Monografia_Rodrigo_Cezar.pdf

https://repositorio.usp.br/directbitstream/296fdd80-e378-45fd-9389-d7753cde5c83/Joel%20Sanvezzo%20Junior_Monografica_Completa_VERS%C3%A3O-FINAL-Joel_Sanvezzo-Redes%20Neurais%20Convolucionais%20Aplicadas%20%C3%A0%20Detec%C3%A7%C3%A3o%20de%20N%C3%A3o-Conformidades%20em%20Equipamentos%20Industriais_206309.pdf

<https://www.ibm.com/br-pt/topics/convolutional-neural-networks>

<https://medium.com/data-hackers/uma-introdu%C3%A7%C3%A3o-as-redes-neurais-convolucionais-utilizando-o-keras-41ee8dcc033e>

<https://paperswithcode.com/dataset/cufsf>