

TRABALHO BIMESTRAL

Comparando diferentes tipos de arquitetura CNN

Guilherme de Matos Oliani
Estudante de Ciência da Computação
Universidade Positivo
Curitiba,Paraná
guilhermeoliani5@gmail.com

1. Introdução

Neste artigo é feita a comparação entre 3 códigos de redes convolucionais, onde, o primeiro intitulado de 'Arch' não utiliza nenhum tipo de arquitetura pronta apenas alguns métodos de classificação e tratamento para as imagens. O segundo código utiliza a arquitetura LeNet que é umas das primeiras CNNs publicadas e por último o terceiro código que é uma arquitetura chamada VGG que é uma arquitetura amplamente reconhecida. O dataset foi utilizado nos três códigos para treinamento.

2. Arquiteturas

2.1 Arch

No código foram utilizadas as bibliotecas como numpy, keras e cv2, foi definida o nome das classes das frutas que no total foram 10, logo depois foi criado o modelo utilizando a classe `Sequential()` do keras, essa classe permite empilhar as camadas em uma sequência para criar a arquitetura do modelo. Após a criação do modelo foi adicionado as camadas convolucionais usando a função `Conv2D()`, sendo que cada camada possui seus hiper parâmetros como o numero de filtros(64,128,256), o tamanho do kernel(3x3), a função de ativação('relu') e a entrada da primeira camada(64,64,3), assim definindo o modelo como uma imagem de 64x64 com 3 canais de cores(RGB).

A cada par de camadas convolucionais é adicionado uma camada de pooling reduzindo a dimensão espacial do que as camadas estão extraindo e permitindo que o modelo aprenda características invariantes. Foi também adicionado após cada camada de pooling uma camada de dropout para prevenir o overfitting. Essas características são então achatadas em um vetor unidimensional utilizando a camada `Flatten()` e alimentadas em camadas totalmente conectadas. As camadas totalmente conectadas são as que irão fazer a classificação final com base na camadas convolucionais anteriores.

Após as camadas conectadas foi utilizado um otimizador chamado SGD onde ele atualiza os pesos baseado na função de perda que em seguida é compilada utilizando o `model.compile` com o modelo definido e o otimizador. É então utilizado o `ImageDataGenerator` que permite diversificar e gerar variações dos dados de treinamento, então os dados são treinados separados em treino, teste e variação e feito o treinamento do modelos.

2.2 LeNet

Esta arquitetura é baseado na arquitetura LeNet onde consiste em duas partes: (i) um codificador convolucional que consiste em duas camadas convolucionais; e (ii) um bloco denso que consiste em três camadas totalmente conectadas. Não foi possível implementar a LeNet exatamente porém foi utilizado as camadas de convolação, Pooling e a função de Relu, além de os dados terem ser tradados com mais eficácia fazendo com que tenha resultados muito melhores que na primeira arquitetura.

2.3 VGG

Esta arquitetura VGG consiste em uma sequência de camadas convolucionais, seguido por uma camada de *pooling* máxima para *downsampling* espacial. A arquitetura VGG é caracterizada por ter camadas convolucionais com filtros de tamanho 3x3 e strides de tamanho 1, seguidas por camadas de pooling com filtros de tamanho 2x2 e strides de tamanho 2. Além disso, a VGG é conhecida por ter várias camadas convolucionais empilhadas antes de realizar o pooling , além de usar as camadas dense para a classificação final.

3. Resultados

- O Arch teve um resultado ruim, onde obteve com 10 épocas 20% de acurácia e em 20 épocas 50%, e no teste com uma nova imagem ele errou o resultado, além da demora em treinar os modelos que passava de 40 minutos.
- A arquitetura baseada em Lenet foi a que melhor, com 27 épocas teve uma acurácia de 100% assim acertando o predict corretamente além de ser muito rápida para treinamento.
- A arquitetura VGG foi a segunda melhor com 45% de acurácia com 2 épocas apenas demorando para treinar em torno de 27 minutos, e acertou o predict.

4. Conclusão

Após análise das arquiteturas o modelo LeNet e VGG foram os melhores para este tipo de dataset, apesar de não ter sido usado as arquiteturas verdadeiras este artigo teve como objetivo demonstrar a diferença de arquiteturas e hiper parâmetros podem gerar no resultado final além de como o pré-processamento das imagens fazem diferença em relação ao modelo Arch. Algumas das diferenças por exemplo entre o LenNet e o VGG são os filtros em cada camada convulacional, o modelo VGG utiliza 64,128,256 já o LeNet utiliza 6 e 16 fltros,o tamanho dos filtros também no VGG é utilizado 3,3 e no LeNet é utilizado 5,5, isso se repete nas camadas de Pooling, camadas densas, função de otimização.

Fontes:

https://pt.d2l.ai/chapter_convolutional-neural-networks/lenet.html

https://pt.d2l.ai/chapter_convolutional-modern/vgg.html

https://www.youtube.com/watch?v=PiF0l6xif-k&t=756s&ab_channel=ShriramVasudevan

https://www.youtube.com/watch?v=bEsRLXY7GCo&ab_channel=ShriramVasudevan