

Relatório CARD 15 - Prática: Redes Neurais Convolucionais 2 (Deep Learning) (II)

Willian Augusto Soder de Souza

O objetivo deste relatório é descrever os principais conhecimentos adquiridos ao assistir às seções 5, 6, 7, 8, 9, 10 e 11 do curso 'Deep Learning: Convolutional Neural Networks in Python', no qual o apresentador se concentra em explicar o funcionamento de redes neurais convolucionais e seus conceitos de forma teórica, com ênfase na parte matemática dos processos. No entanto, ele também demonstra a implementação prática em Python utilizando a biblioteca TensorFlow, além de trabalhar com alguns datasets, como o CIFAR-10. Abaixo segue um resumo dos principais conceitos teóricos apresentados nessas seções.

- **Convolução:** é o processo que permite a uma rede neural convolucional identificar e aprender características importantes de uma imagem, desempenhando um papel central na capacidade dessas redes de realizar tarefas como classificação de imagens e detecção de objetos.
- **Stride:** é um parâmetro crucial que determina como o filtro se desloca pela imagem durante a convolução, afetando diretamente o tamanho e a resolução do mapa de características resultante. É uma ferramenta poderosa para controlar o equilíbrio entre a preservação de detalhes finos e a redução da dimensionalidade dos dados.
- **Filtro (Kernel):** é uma pequena matriz de números (também chamados de pesos) que é usada para extrair características específicas de uma imagem ou outro tipo de dado, ele funciona “deslizando” sobre a imagem realizando operações de soma e multiplicação dos valores dos pixels. Em uma CNN, os filtros são aprendidos durante o processo de treinamento para identificar padrões importantes nos dados de entrada.
- **Padding:** refere-se ao processo de adicionar pixels ao redor da borda de uma imagem antes de aplicar a operação de convolução. Isso é feito para controlar o tamanho do mapa de características resultante e para garantir que as características nas bordas da imagem sejam consideradas na mesma medida que as características no centro.
- **Mapa de características:** é um componente crucial em redes neurais convolucionais, representando as informações extraídas de uma imagem após a aplicação de filtros (ou kernels) durante o processo de convolução. Ele contém as respostas dos filtros aplicados à imagem de entrada, destacando padrões específicos, como bordas, texturas e outras características relevantes.
- **Redes neurais convolucionais (CNNs):** é um tipo de rede neural projetada especificamente para processar dados com uma estrutura de grade, como imagens. Ela é composta por várias camadas que automaticamente aprendem a extrair características relevantes dos dados de entrada para realizar tarefas como classificação, reconhecimento de padrões, e detecção de objetos.

Componentes principais:

- **Camadas Convolucionais:** Aplicam filtros (kernels) à entrada para extrair características, como bordas, texturas, e formas. Cada filtro gera um mapa de características, que captura diferentes aspectos da imagem.

- **Camadas de Pooling:** Reduzem a dimensionalidade dos mapas de características, preservando as informações mais importantes, o que ajuda a tornar o modelo mais eficiente e robusto a variações na imagem.
 - **Flattening:** é o processo de converter um tensor multidimensional, como um mapa de características gerado pelas camadas convolucionais e de pooling, em um vetor unidimensional.
 - **Camadas Totalmente Conectadas:** Recebem os mapas de características processados e os utilizam para tomar decisões finais, como classificar uma imagem em uma categoria específica.
- **Gradient Descent:** é um algoritmo de otimização utilizado em machine learning e redes neurais para ajustar os parâmetros de um modelo (como os pesos em uma rede neural) de forma a minimizar uma função de custo ou perda. O objetivo é encontrar os valores ótimos dos parâmetros que reduzem ao máximo o erro nas previsões do modelo.
 - **Stochastic Gradient Descent:** é uma variação do algoritmo de otimização Gradient Descent, onde a principal diferença entre o SGD e o GD tradicional está na forma como os gradientes são calculados e utilizados para atualizar os parâmetros do modelo.
 - **Momentum:** é uma técnica de otimização usada em redes neurais para acelerar o treinamento e melhorar a convergência dos algoritmos de descida de gradiente, como o Gradient Descent. O conceito de Momentum é inspirado na física, onde um objeto em movimento continua a se mover devido à inércia. Na otimização, o Momentum ajuda a "acelerar" a descida ao longo das direções onde o gradiente é consistentemente grande e a "suavizar" as atualizações em direções onde o gradiente muda rapidamente.
 - **Variable and Adaptive Learning Rates:** são técnicas usadas para ajustar a taxa de aprendizado durante o treinamento de modelos.
 - **Adam:** é um algoritmo de otimização que ajusta a taxa de aprendizado automaticamente para cada parâmetro de um modelo, combinando a média móvel dos gradientes e dos quadrados dos gradientes. Isso ajuda a acelerar a convergência e a estabilizar o treinamento.

CONCLUSÃO:

O aprendizado dos conceitos teóricos e a compreensão detalhada da matemática envolvida nos processos de redes neurais convolucionais são fundamentais para o desenvolvimento e a aplicação eficaz desses modelos em tarefas complexas de machine learning. A partir da análise das técnicas como convolução, pooling, flattening e otimização, bem como dos algoritmos avançados de otimização como o Adam e suas variantes, é evidente que uma base sólida na teoria matemática é crucial para um entendimento profundo e uma implementação bem-sucedida.

Os conceitos de convolução e pooling, por exemplo, não apenas descrevem como as redes convolucionais extraem e reduzem características, mas também explicam como essas operações afetam o desempenho e a capacidade do modelo. O entendimento detalhado da etapa de flattening e dos mapas de características permite a aplicação correta dos dados processados em camadas totalmente conectadas, impactando diretamente a eficácia da classificação e outras tarefas de previsão.