

Relatório LAMIA 15

Prática: Redes Neurais Convolucionais 2 (Deep Learning) (II)

Kaique Medeiros Lima

1 Introdução

Neste card, é apresentado o conteúdo de Redes Neurais Convolucionais, comumente utilizada em tarefas de processamento de imagens e também, de textos. Foi utilizado o conteúdo do curso *Deep Learning: Convolutional Neural Networks in Python* da Udemey.

2 Descrição da atividade

2.1 Convolução

A convolução é uma operação matemática fundamental em redes neurais, utilizada nas *Convolutional Neural Networks* (CNNs). É uma forma de combinar duas funções, resultando em uma terceira, sendo particularmente aplicada no processamento de imagens. A convolução permite extrair características importantes, como bordas, texturas e objetos, funcionando como um “modificador” que destaca padrões relevantes nas imagens.

2.1.1 Kernel ou Filtro

Durante a convolução, a matriz de entrada (imagem) é multiplicada por uma matriz menor chamada *kernel* ou filtro. Esse kernel tem tamanhos pré-definidos, como 3×3 , 5×5 ou 7×7 . A operação de convolução é realizada movendo o filtro sobre a matriz de entrada e multiplicando seus elementos, gerando uma matriz de saída menor que contém as informações extraídas.

2.1.2 Stride

O *stride* define o número de pixels que o filtro “anda” a cada iteração. Um *stride* maior significa que mais pixels são desconsiderados, resultando em uma matriz de saída ainda menor. Strides maiores podem ser úteis para reduzir o tamanho da imagem processada sem perder muita informação.

2.1.3 Padding

O *padding* consiste em adicionar pixels com valor zero nas bordas da imagem de entrada, garantindo que o tamanho da imagem original seja mantido após a aplicação do filtro. Ele também ajuda a evitar a perda de informações nas extremidades da imagem durante a convolução.

2.1.4 Mapa de Características

O mapa de características é o resultado gerado após a aplicação dos filtros. Cada filtro extrai uma característica diferente da imagem (como bordas ou texturas), gerando múltiplos mapas de características, que representam as informações obtidas de acordo com o filtro utilizado.

2.2 Data Augmentation

Como precisamos da maior variabilidade de dados possíveis para o treinamento de um modelo eficiente, o data augmentation é uma série de pequenas modificações nas imagens, como ampliar, virar de ponta cabeça ou girar 90 graus, fazendo com que os dados tenham sua variação aumentada, por isso o nome, data augmentation.

2.3 Batch Normalization

Os dados utilizados para o treinamento de um modelo normalmente sofrem do problema da normalização dos dados enquanto passam pelas camadas, perdendo sua normalização, por isso existe o batch normalization, o qual normaliza os dados após cada camada para o melhor aperfeiçoamento do modelo.

2.4 Embeddings

É uma técnica usada para converter dados categóricos, como palavras ou itens, em vetores de dimensões mais baixas mas de mesma qualidade, que capturam suas relações semânticas. Essa representação permite que modelos processem dados de forma eficiente, mantendo proximidade entre elementos com significados ou contextos semelhantes.

2.5 Redes Neurais Convolucionais (CNNs)

As Redes Neurais Convolucionais (CNNs) são compostas por várias camadas, cada uma com a função de manipular e extrair informações das imagens de entrada. Essas redes são projetadas especificamente para trabalhar com dados de imagens e são amplamente utilizadas em tarefas como classificação de imagens, detecção de objetos e reconhecimento facial. Nas CNNs, os filtros aplicados nas camadas convolucionais extraem padrões visuais, como bordas, formas e texturas, em diferentes níveis de abstração. Essas características são, então, passadas para as camadas densas, onde as informações são combinadas e processadas para realizar a tarefa de classificação.

2.6 Convolução em imagens 3D

A convolução em imagens 3D é uma extensão da convolução 2D, usada para processar dados 3D. Nessa abordagem, um kernel se desloca por todos os pixels, capturando relações espaciais entre eles. Esse processo permite que ao modelo extraia características complexas ao longo de todo o volume da imagem, sendo essencial para análises de dados 3D, onde a profundidade é um fator relevante.

2.7 Tracking shapes in a CNN

Cada camada aplica filtros que capturam diferentes características, como bordas, texturas e formas mais complexas. À medida que a imagem passa por essas camadas, a CNN aprende a identificar partes relevantes da forma, independentemente de sua posição, tamanho ou orientação. A convolução sempre funciona, independente do tamanho da camada de entrada pois é sempre calculado conforme seu tamanho, sendo diretamente proporcional.

2.8 Estrutura de uma CNN

2.8.1 Camada de Convolução

A camada de convolução é responsável pela extração de características das imagens. Ela aplica filtros sobre a imagem de entrada, gerando o mapa de características. Além disso, técnicas de normalização, como *Batch Normalization*, são frequentemente aplicadas nesta etapa para acelerar o treinamento e melhorar a estabilidade do modelo.

2.8.2 Camada de Pooling

A camada de *pooling* é usada para reduzir a dimensionalidade do mapa de características, o que diminui a complexidade do modelo e melhora a eficiência computacional, preservando as informações mais importantes. As técnicas mais comuns de pooling são o *Max Pooling*, que seleciona o valor máximo de cada região do mapa, e o *Average Pooling*, que calcula a média dos valores.

2.8.3 Flattening

O *flattening* transforma os dados multidimensionais gerados pelas camadas anteriores em um vetor unidimensional, necessário para a entrada nas camadas densas. Essa operação é crucial, pois as camadas densas requerem uma representação unidimensional dos dados para funcionar corretamente. Alternativamente, o *Global Max Pooling* (GMP) pode ser usado, que reduz cada mapa de características a um único valor representando a característica mais forte. A principal diferença entre o *flattening* e o GMP é que o primeiro preserva todas as características aprendidas, enquanto o segundo sintetiza as informações em valores mais compactos e representativos.

2.8.4 Camadas Densas

As camadas densas são compostas por neurônios completamente conectados aos neurônios da próxima camada. É nelas que ocorre o processamento final das características extraídas anteriormente, e onde a classificação ou predição é realizada. Essas camadas combinam as informações extraídas das imagens para fornecer o resultado final do modelo.

2.9 Aprofundando: Funções de perda

2.9.1 Mean Squared Error (MSE)

É normalmente usada em problemas de regressão. Ela mede a diferença média quadrática entre as previsões do modelo e os valores reais. Como o erro é elevado ao quadrado, essa

função penaliza erros maiores de forma mais intensa, incentivando o modelo a ajustar suas previsões para minimizar discrepâncias grandes.

2.9.2 Binary Cross-Entropy

É usada em problemas de classificação binária, onde a saída é uma probabilidade entre 0 e 1. O BCE penaliza previsões incorretas mais severamente quando o modelo está muito confiante no resultado errado, sendo ideal para problemas como detecção de fraudes ou classificação de imagens binárias.

2.9.3 Categorical Cross-Entropy

É usada para problemas de classificação com múltiplas classes (não binários). Ela compara a distribuição verdadeira dos rótulos categóricos com as probabilidades preditas para cada classe. O objetivo é minimizar a diferença entre a classe real e as previsões, incentivando o modelo a atribuir maior probabilidade para a classe correta.

2.10 Aprofundando: Gradient Descent e Otimizadores

2.10.1 Gradient Descent

É um algoritmo de otimização amplamente usado para minimizar funções de custo em redes neurais. Ele funciona ajustando os parâmetros do modelo em direção ao gradiente descendente da função de perda, ou seja, na direção que reduz o erro. Embora eficaz, o gradient descent clássico requer que o cálculo do gradiente seja feito em todo o conjunto de dados, o que pode ser ineficiente para grandes volumes de dados.

2.10.2 Stochastic Gradient Descent

É uma variação do gradient descent em que os parâmetros do modelo são atualizados por um conjunto de amostras, em vez de usar todo o conjunto de dados. Isso torna o processo mais rápido e eficiente em grandes conjuntos, mas também introduz mais variação nas atualizações, o que pode levar a oscilações na função de perda.

2.10.3 Momentum

É uma técnica utilizada em conjunto com o gradient descent para suavizar a convergência, especialmente em problemas com gradientes oscilantes. Ele funciona adicionando um termo que acumula o histórico de gradientes passados, permitindo que o modelo mantenha sua direção quando o gradiente muda de forma inconsistente. O momentum age como uma memória do gradiente, suavizando as atualizações e permitindo um avanço mais rápido em direção ao mínimo global.

2.10.4 Adaptive Learning Rates

Os métodos de Adaptive Learning Rates ajustam dinamicamente a taxa de aprendizado para cada parâmetro, em vez de usar um valor fixo. O step decay derruba a learning rate após algumas épocas, o exponential decay faz a mesma coisa, mas diferente do step que possui sua curva como uma escada, o exponential mantém uma curva exponencial.

2.10.5 Adam

É uma técnica para a modificação da descida do gradiente, sendo o método de escolha para a maioria das redes neurais atuais. É um algoritmo de otimização que combina as ideias de momentum e adaptive learning rates.

3 Conclusão

Neste card, foi apresentado o conceito de convolução e Redes Neurais Convolucionais (CNNs), destacando a importância dessas técnicas no processamento de imagens e reconhecimento de padrões. As CNNs são amplamente utilizadas em tarefas de visão computacional, como classificação de imagens, devido à sua capacidade de extrair características relevantes das imagens e realizar tarefas complexas de forma eficiente. Também foram abordados conceitos como data augmentation, batch normalization, embeddings e funções de perda, que são fundamentais para o treinamento e otimização de modelos de deep learning.