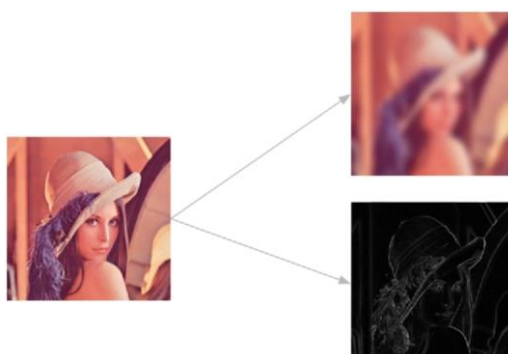


Convolutional Neural Networks

O que é uma CNN?

De forma breve, uma rede neural convolucional é uma "rede neural com convolução", e para entendermos melhor essa rede neural precisamos saber o que é a convolução.

Por começo podemos considerar a convolução como um filtro de imagem, apontado no vídeo dois exemplos sendo eles a convolução para o borrão da imagem e para a detecção de bordas.



A convolução no esquema geral é a simples tarefa de somar e multiplicar para chegar nos filtros desejados, imagine uma imagem 3x3 multiplicando por um filtro (kernel) 2x2. Considere que estamos somando a multiplicação das matrizes para achar o resultado reduzido.

Input		Kernel		Output																	
<table border="1"><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td>8</td></tr></table>	0	1	2	3	4	5	6	7	8	*	<table border="1"><tr><td>0</td><td>1</td></tr><tr><td>2</td><td>3</td></tr></table>	0	1	2	3	=	<table border="1"><tr><td>19</td><td>25</td></tr><tr><td>37</td><td>43</td></tr></table>	19	25	37	43
0	1	2																			
3	4	5																			
6	7	8																			
0	1																				
2	3																				
19	25																				
37	43																				

O que realmente é feito?

No deep learning, a real função desta rede neural é a correlação cruzada, na qual apresenta a mesma função da convolução, porém com um grau facilitado de compreensão.

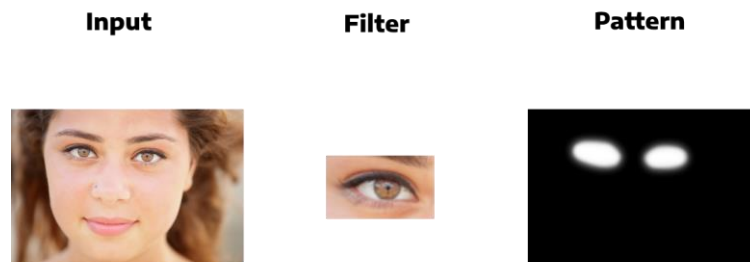
A primeira instância, convolução não possui um significado direto, diferente da correlação, onde podemos considerar que algo correlacionado são "parecidos".

X e Y são correlacionados porque possuem um grau de similaridade entre eles.

Pense na rede neural convolucional como uma "rede neural correlacional".

Uma outra perspectiva sobre convolução

Podemos utilizar a convolução para encontrar padrões em imagens, utilizando cálculos matemáticos como o produto escalar para analisar esses padrões e confirmar caso eles se repitam no restante da imagem.



Outra utilidade para a convolução é a facilidade e a qualidade ao comparar com outros métodos, por exemplo a multiplicação direta de matrizes, na qual pode se tornar algo complexo e custoso para o computador, mas ao utilizar da convolução se torna um processo "barato computacionalmente".

Por que usar o índice 0?

Ao utilizar a variante de deep learning, o índice começando em 0 ajuda a "organizar" melhor os cálculos, resultando a mesma resposta em ambos os casos, o índice 0 mantém um padrão visualmente facilitado.

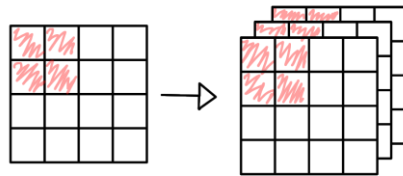
Ao considerar a variante matemática, o fato de utilizar o índice 0 como 1 causa problemas de posicionamento e lógica, considerando que as multiplicações precisam passar por diversas etapas, deixando assim uma maior chance para acontecer erros e complicando a lógica do processo.

Convolução de imagens coloridas

O conceito da convolução quando aplicada a imagens coloridas não difere tanto do já aprendido.

Sabemos que uma imagem colorida é representada por 3 dimensões, de forma breve a matriz filtro passará pela terceira camada inteira e seguirá o mesmo padrão anterior com as outras duas.

Imaginando sua função de encontrar padrões, ao utilizar uma imagem colorida é ampliado a sua "dificuldade" ao considerar que esses padrões precisam possuir cores próximas.

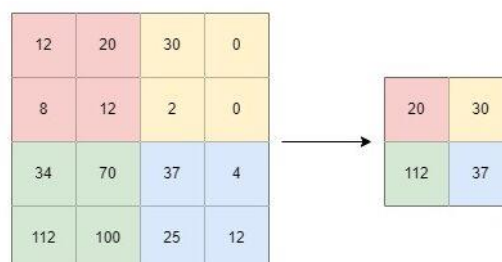


Ao considerar imagens coloridas, o uso da rede neural convolucional é de extrema importância considerando métodos como a multiplicação de matriz, considerando uma imagem 32x32x3 e um vetor de saída 28x28x64, a multiplicação direta chega a possuir 32 mil parâmetros a mais que a CNN.

Arquitetura da CNN

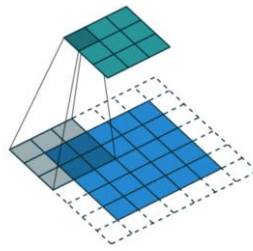
Inicialmente classificando como uma CNN típica funciona, podemos separar em dois estágios, sendo o primeiro um “achador de características”, e o segundo um classificador não-linear para essas características.

Neste primeiro estágio, temos a convolução e um processo de Pooling, na qual é responsável por descartar uma parte dos dados a fim de reduzir a quantidade de dados a serem processados.

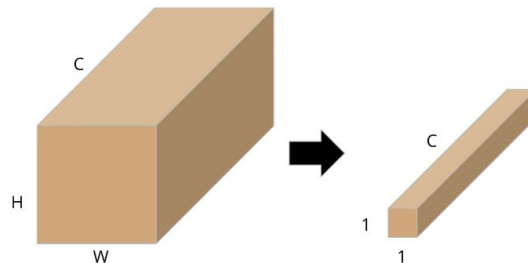


A razão para o Pooling acontecer logo após uma iteração da convolução é a de diminuir a imagem a ser analisada, e considerando que o filtro mantém o tamanho inicial, facilitando assim encontrar padrões maiores.

Uma alternativa ao Pooling é a convolução Strided, na qual "chutamos" um valor para os pixels ao redor, considerando que a probabilidade de serem parecidas ou próximas ao centro é bastante alta.



Já no segundo estágio, a rede neural densa é responsável por modificar o formato da imagem para deixar de forma apropriada, normalmente sendo utilizado a função Flatten() a fim de um resultado de uma dimensão.



Preparação do código para a CNN

- Passo 1 – Carregar os dados: Normalmente em padrão MNIST e/ou CIFAR-10, para imagens preto e branco e coloridas respectivamente.
- Passo 2 – Construir o modelo: Utilizando a rede neural convolucional para uma melhora nos resultados esperados. Também é possível ser utilizado uma API funcional, facilitando assim na criação dos modelos.
- Passo 3 – Treinar o modelo: Utilizar dos dados presentes para treinar o modelo, fazendo com que ele identifique padrões que retornam certos resultados.
- Passo 4 – Aprimorar o modelo: Utilizando métodos para melhorar a precisão final.
- Passo 5 – Fazer previsões: Utilizar o modelo treinado para prever possíveis resultados, tanto utilizando os mesmos dados do treinamento quanto novos para uma melhor visão da eficiência deste modelo.

Aprimorar os dados

Utilizado no passo 4 da preparação, o ato de aprimorar os dados é resultado de modificar as imagens/dados presentes de formas que diferenciem entre ambas, porém mantenham a essência da imagem.

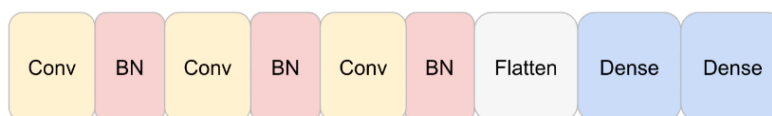
Por exemplo, considere a imagem de um cachorro, podemos inverter ela, tirar as cores, aproximar, borrar, entre outras formas.



Normalização do lote

Os dados da CNN são normalizados ao "entrar" na rede neural, porém eles podem acabar perdendo essa normalização conforme o processo, para evitar este comportamento é utilizado o Batch Normalization.

Este método de normalização normalmente é colocado após cada convolução, mesmo com sua utilidade para manter os dados normalizados, nem sempre o resultado é positivo, podendo apresentar problemas e quedas na precisão.



Lidando com dados em formato de texto

Um problema encontrado ao tratar destas redes neurais é o uso de palavras como dados para o treinamento/teste, utilizando a fórmula para encontrar estes padrões se torna impossível ao considerar uma palavra ao invés de um valor.

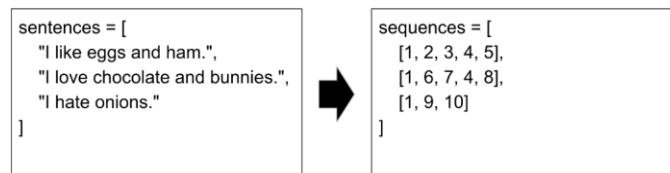
Uma possível solução é o One-Hot Encoded já mencionado anteriormente, na qual transformamos a palavra para um formato de vetor na qual é composto majoritariamente de zeros com o valor 1 adicionado para a posição respectiva da palavra.

Por exemplo, considere a frase: "Eu amo o Lamia", podemos separar em 4 palavras e considerar o vetor respectivo como:

"Eu": [0, 0, 0, 1], "amo": [0, 0, 1, 0], "o": [0, 1, 0, 0], "Lamia": [1, 0, 0, 0].

Porém existe um grande problema com este método, ao considerar grandes textos com diversas possíveis palavras, o tamanho do vetor se torna massivamente custoso e complexo de se operar.

A solução para este problema, utilizar um tratamento chamado de Embedding. Começamos convertendo as palavras em inteiros distintos, depois adicionamos um vetor respectivo para cada inteiro, atribuindo assim um peso para ajudar a distanciar eles.

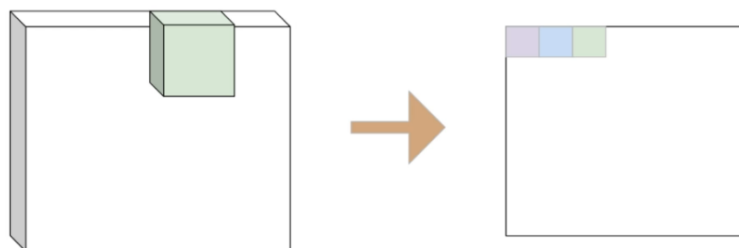


O seu processo de convolução se torna simples ao considerar que contém apenas uma dimensão, fazendo de forma direta a multiplicação de uma linha grande com uma linha menor.

Convolução para 3 dimensões

O produto com 3 dimensões mais comum de ser utilizado com a convolução são as imagens coloridas, onde a terceira camada normalmente será das suas cores RGB.

Considerando a diferença para as convoluções de 1 e 2 dimensões, ela possui um método semelhante de ser utilizado, passando cada quadrante simultaneamente nas três camadas extras. O problema se encontra no fato da convolução com entrada tridimensional retornar uma saída bidimensional.

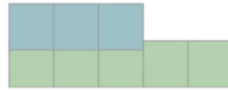


A solução é passar um filtro que utiliza 4 dimensões, de forma que duplicamos a camada responsável pelo filtro. Por exemplo, considere um filtro de detecção de bordas, adicionamos uma multiplicação extra com o mesmo filtro para gerar essa quarta dimensão, transformando assim o resultado para 3 dimensões.

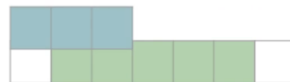
Tipos de convolução

Durante a camada de convolução, existem três tipos que são utilizados para gerar o resultado esperado, o "Valid", "Same" e o "Full".

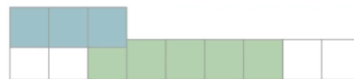
Valid: O filtro nunca sai da entrada.



Same: O resultado esperado deve ser de mesmo tamanho da entrada.



Full: O filtro é permitido sair da entrada o suficiente para que sobre apenas 1 quadrado da entrada.



Padrões para criar uma CNN

Conforme aprendemos mais sobre esse tipo de rede neural, percebemos que existem diversas variáveis nas quais podemos escolher como utilizar, isso faz com que os padrões normalmente utilizados se tornem importantes para seguir uma base.

O primeiro padrão a ser seguido, a rede neural deve consistir em duas partes, a primeira sendo uma série de convoluções e a segunda uma série de camadas completamente conectadas.

Outro padrão a ser seguido é considerar que cada camada é composta por uma imagem na qual possui padrões a serem "descobertos".

A série de camadas conectadas normalmente é criada com o mesmo ou menor número de camadas ocultas (mesmo tamanho sendo o mais comum). Manter a quantidade de unidades escondidas também facilita a implementação.

Funções de perda

Mean Squared Error (MSE): Uma função de perda usada principalmente em problemas de regressão. Ela calcula a média dos quadrados das diferenças entre os valores previstos pelo modelo e os valores reais.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Binary Cross Entropy (BCE): Uma função de perda usada para problemas de classificação binária, onde o objetivo é classificar as entradas em uma de duas classes. Ela mede a diferença entre a distribuição real dos rótulos e a distribuição prevista pelo modelo.

$$Loss = -\frac{1}{N} \sum_{i=1}^N \{y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)\}$$

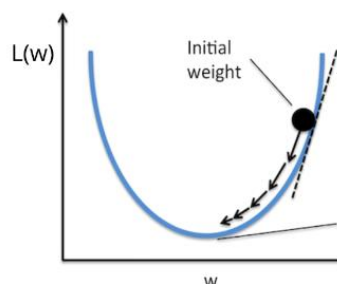
Categorical Cross Entropy (CCE): Uma função de perda usada para problemas de classificação multi-classe, onde o objetivo é classificar as entradas em uma de várias classes. Ela mede a diferença entre a distribuição real dos rótulos e a distribuição prevista pelo modelo para cada classe.

$$Categorical\ Cross - Entropy = - \sum_{i=1}^N \sum_{k=1}^K y_{ik} \log \hat{y}_{ik}$$

Método do gradiente

É o método utilizado para treinar todos os modelos, é implementado em neurônios, redes neurais e além. Considerado como o fundamento que liga todos esses sistemas no deep learning.

A ideia deste método é simples, porém efetiva, considere pequenos passos por iteração, cada passo dado ele se aproxima do menor erro possível, do lugar mais baixo do "gráfico", eventualmente ele chegará no lugar mais baixo possível.



Método do gradiente estocástico

O que diferencia este método do anterior é a capacidade de se adaptar a quantidades de dados imensos.

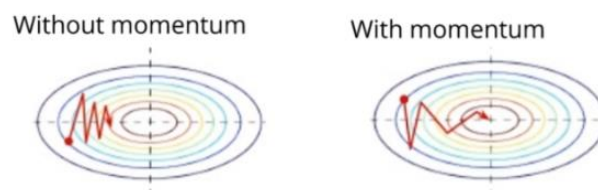
Considere por exemplo a população mundial como 7 bilhões de pessoas, seria impossível a tarefa de perguntar individualmente alguma pergunta específica. Utilizando o método estocástico, pegamos uma pequena parcela desses 7 bilhões de forma aleatória e consideramos como uma aproximação do resultado.



Momento

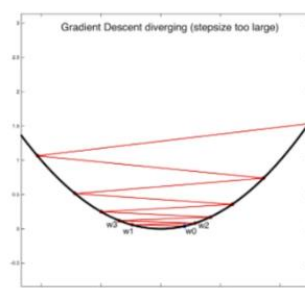
Uma tática a ser utilizada quando consideramos o método do gradiente é o uso do momento para evitar grandes períodos de iterações com pouco avanço.

O momento pode ser aplicado como um multiplicador por iteração para ir modificando a força que ele move pelo gradiente e se adaptar conforme procede o treinamento.



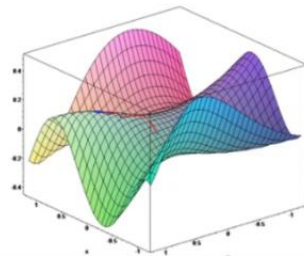
Taxa de aprendizagem adaptável e variável

A taxa de aprendizagem variável possui a intenção de diminuir o tamanho de cada "passo" após certa quantidade de iterações. Imagine que você está próximo do valor mínimo possível, caso de um passo muito grande ficará num loop sem conseguir chegar no real valor mínimo.



A taxa de aprendizagem adaptável como o próprio nome já diz, se adapta conforme cada passo é tomado, considere por exemplo que um passo a esquerda seja uma queda e a

direita seja uma planície, o passo para a esquerda deve ser menor do que o oposto. Ele é aplicado em alguns métodos, sendo os principais AdaGrad e o RMSProp.

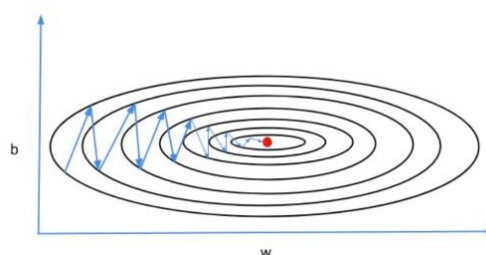


Adam (Adaptive Moment Estimation)

Considerado o método principal para otimizar redes neurais, normalmente utilizado como escolha padrão na criação da rede, é um método muito eficiente ao utilizar suas configurações padrões (learning rate por exemplo).

Podemos visualizar o Adam como um RMSProp, porém utilizando momento. O Adam combina as vantagens do AdaGrad, que trabalha bem com gradientes esparsos, e do RMSProp, que é robusto ao ruído. Ele calcula estimativas adaptativas de momentos de primeira e segunda ordem, ajustando o learning rate para cada parâmetro com base nas médias móveis das primeiras e segundas derivadas dos gradientes. Isso permite que o Adam se adapte de forma mais eficaz às mudanças na função de perda, proporcionando uma convergência rápida e estável.

Em resumo, o Adam otimiza o processo de treinamento ajustando dinamicamente a taxa de aprendizado de cada parâmetro com base em informações coletadas durante o treinamento, o que o torna uma escolha eficiente e popular para uma ampla gama de problemas de aprendizado profundo.



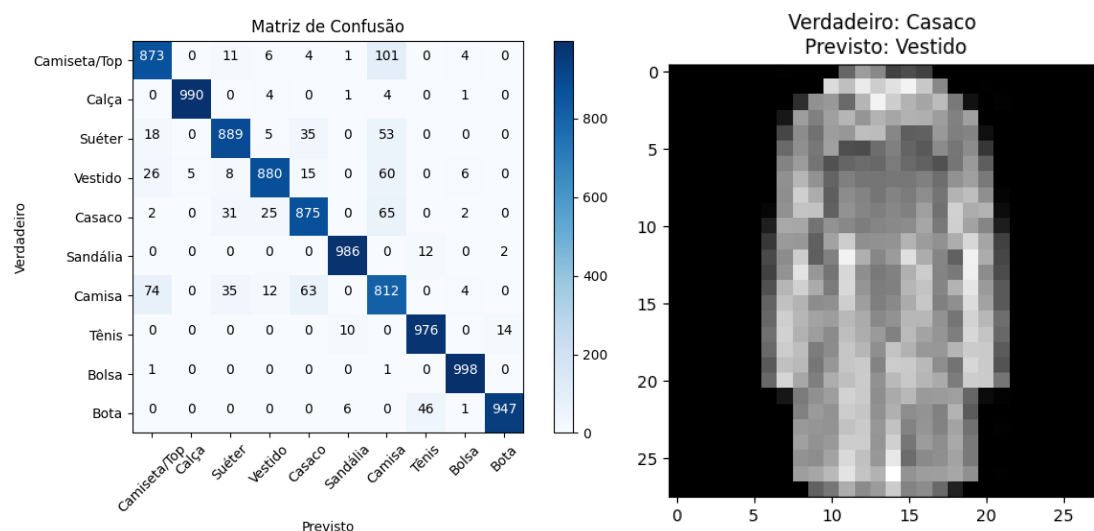
Atividade

Durante o curso, diversas atividades práticas são realizadas para proporcionar uma compreensão mais profunda dos conceitos de redes neurais convolucionais e sua aplicação em problemas reais. As atividades incluem uma implementação do Fashion

MNIST utilizando de conceitos aprendidos durante as aulas, um aprimoramento da rede neural gerada e dois códigos responsáveis por tratar de textos na rede CNN.

Atividade Criada

Neste projeto, foi desenvolvido um modelo de rede neural convolucional (CNN) para classificar imagens de roupas usando a base de dados Fashion MNIST. A atividade incluiu o pré-processamento dos dados, construção e treinamento do modelo CNN, e avaliação do desempenho com a geração de uma matriz de confusão. Adicionalmente, foram implementadas funções para salvar e carregar o modelo, bem como para exibir exemplos de classificações incorretas. Este projeto aprimora o que foi mostrado no curso utilizando técnicas para gerar uma acurácia melhor e possui um modo de organização demonstrado em cards anteriores.



Conclusão

Este relatório explorou a aplicação de redes neurais convolucionais (CNNs) no reconhecimento de padrões em imagens. A CNN é uma ferramenta poderosa para analisar dados visuais, utilizando convoluções para identificar e classificar padrões complexos em imagens. O projeto desenvolvido demonstrou a aplicação prática dessas técnicas na tarefa de classificar imagens de roupas, abordando desde o pré-processamento dos dados até a avaliação do modelo.