

Implementando a estrutura de uma Rede Neural Convolucional utilizando o MapReduce do Spark

Rafael Guimarães Sakurai¹

*Universidade Federal do ABC
Pós-Graduação em Ciência da Computação
Disciplina: Inteligência na Web e Big Data*

Abstract

A classificação de imagens é uma tarefa na qual dada uma figura é obtido uma saída que representa sua classe (rótulo) ou a probabilidade das classes que descrevem a figura. Neste documento é apresentado um exemplo de problema que pode ser tratado por meio de Redes Neurais Convolucionais sendo implementado utilizando a técnica de MapReduce do Spark.

Keywords: MapReduce, Rede Neural Convolucional, Spark

1. Rede Neural Convolucional

A Rede Neural Convolucional (do inglês *Convolutional Neural Network* - CNN) vem ganhando destaque na classificação de imagens. A Figure 1 apresenta um exemplo de arquitetura de CNN.

5 O diferencial das CNNs está nas diversas camadas convolucionais, que aplica uma função matemática de Convolução nos dados de entrada e depois realizando o Agrupamento (*pooling*). A saída da convolução é passada para a próxima camada convolucional até chegar na última camada conhecida como Camada Densa que normalmente é representada por uma rede *Perceptron* de múltiplas
10 camadas (do inglês *Multilayer Perceptron* - MLP).

¹Email: rafasakurai@gmail.com

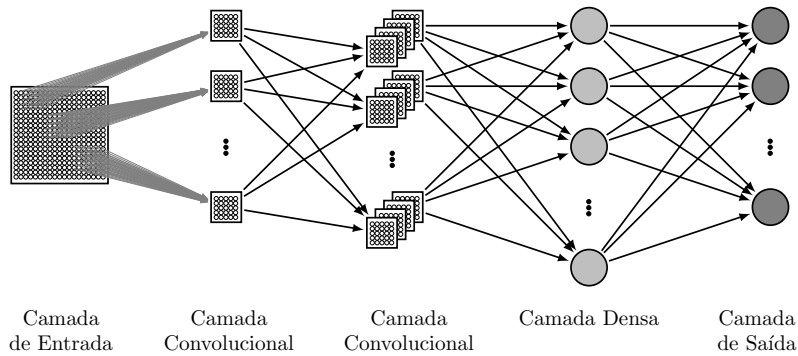


Figure 1: Exemplo de topologia da Rede Neural Convolucional.

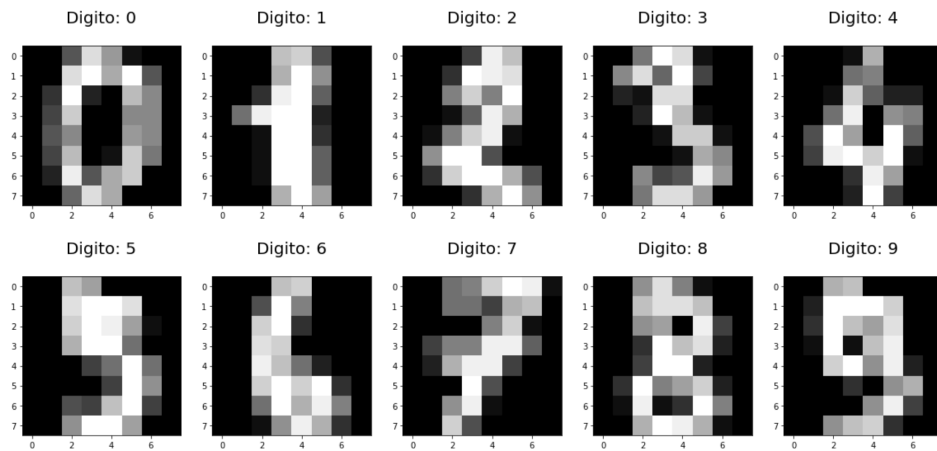


Figure 2: Exemplo dos dígitos utilizados no treinamento.

2. Conjunto de dados de treino

A biblioteca *scikit-learn* possui internamente um conjunto de dados composto por 1.797 exemplos de imagens de dígitos 0 à 9.

Neste conjunto de dados estão disponíveis os dígitos e sua classe, como ap-
 15 resentado na Figure 2. Cada dígito é formado por uma matriz de dimensão de 8×8 , como mostrado na Table 1.

Table 1: Matriz de dimensão 8×8 com os valores que correspondem ao exemplo de um dígito zero.

0.0	0.0	5.0	13.0	9.0	1.0	0.0	0.0
0.0	0.0	13.0	15.0	10.0	15.0	5.0	0.0
0.0	3.0	15.0	2.0	0.0	11.0	8.0	0.0
0.0	4.0	12.0	0.0	0.0	8.0	8.0	0.0
0.0	5.0	8.0	0.0	0.0	9.0	8.0	0.0
0.0	4.0	11.0	0.0	1.0	12.0	7.0	0.0
0.0	2.0	14.0	5.0	10.0	12.0	0.0	0.0
0.0	0.0	6.0	13.0	10.0	0.0	0.0	0.0

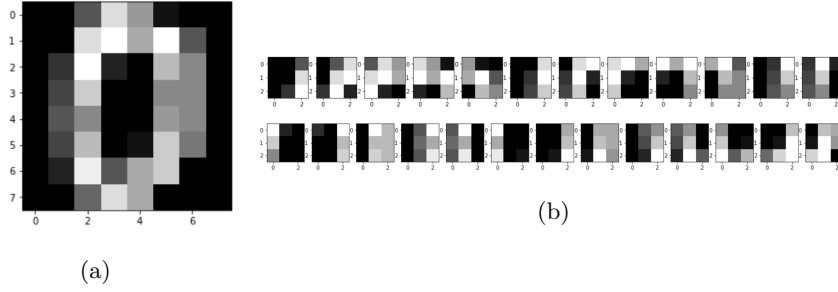


Figure 3: Exemplo de partes que podem ser geradas a partir do dígito zero.

3. Aplicando Redes Neurais Convolucionais para classificação dos dígitos

Neste exemplo será apresentado o uso da CNN para a partir dos dados do
20 dígitos tentar classificá-lo.

3.1. Realizando a convolução

Com base na entrada, por exemplo o dígito zero representado na Figure 3a, pode ser aplicado um mapeamento para gerar diversar partes com tamanho igual ao do *kernel*, como mostrado na Figure 3b.

25 A convolução é aplicada por meio da Equation 1.

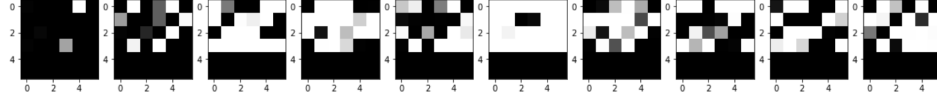


Figure 4: Exemplo de mapa de ativações.

$$H(x, y) = \sum_{a=0}^{n-1} \sum_{b=0}^{n-1} E(x-a, y-b)K(a, b), \quad (1)$$

em que dado uma entrada E e um *kernel* K de dimensão $n \times n$, é somado o resultado da multiplicação de cada posição do *kernel* por uma área correspondente a partir da posição x e y da entrada, obtendo como resultado uma resposta de ativação. Após aplicar o *kernel* em toda área da matriz de entrada, o resultado
30 obtido é um mapa com todas as ativações.

A Figure 4 apresenta um exemplo visual do que poderia ser a representação visual de um mapa de ativações a partir de dez *kernels*.

3.2. Agrupamento com Max Pooling

Após gerado o mapa de ativações é realizado seu agrupamento, normalmente
35 utilizando a função *Max Pooling*, que agrupa regiões do mapa de ativações mantendo apenas o maior valor de cada região, assim gerando um mapa de ativações mais compacto e mantendo sua principais ativações.

3.3. Camada densa com Multilayer Perceptron

A cada camada convolucional serão gerados mais mapas de ativações, a
40 última camada convolucional passará os mapas para uma MLP que gera como saída a probabilidade de cada dígito entre 0 e 9 dada a entrada inicial da CNN.

4. Montando a Rede Neural Convolucional com o MapReduce

A estrutura da CNN é formada por várias camadas que podem ser representadas por meio do mapeamento feito pelo *MapReduce* do Spark.

45 Uma forma de representar uma CNN usando MapReduce seria:

1. Para cada camada convolucional:
 - (a) Mapeia a entrada para uma matriz com várias partes de tamanho igual ao *kernel*;
 - (b) Mapeia cada parte da entrada com o kernel, aplicando a função de ativação e gerando um mapa de ativações;
 - (c) Aplica o agrupamento em cada mapa de ativações gerando como saída um mapa de ativações com dimensão reduzida;
 - (d) Junta todos os mapas de ativações, pois serão passados como entrada para a camada seguinte.
2. A saída da última camada convolucional é passada para uma MLP, que:
 - (a) Mapeia os mapas de ativações para um vetor sequencial;
 - (b) Aplica a função de ativação da MLP gerando como saída dez probabilidades representando cada um dos dígitos.

A implementação deste exemplo está disponível no [GitHub](#).

5. Conclusão

A estrutura da Rede Neural Convolucional pode ser aplicada utilizando a técnica de MapReduce, tendo como objetivo paralelizar o processo de convolução e agrupamento. Mas é necessário obter os mapas de ativações gerados por todos os *kernels* de uma camada convolucional, antes de seguir adiante, pois estes mapas serão a entrada da camada convolucional seguinte. Também é necessário aguardar todos as convoluções paralelas finalizarem para obter a entrada da MLP e após isso, com sua saída, aplicar a retropropagação para ajustar os pesos.