

# Autoencoder: Redimensionalidade de Dados

Caio Ueno, Claudia Sanches

<sup>1</sup>Departamento de Computação  
Universidade Federal de São Carlos (UFSCar)  
São Carlos – SP – Brasil

caioluiggy@hotmail.com, claudia.rincon.sanches@gmail.com

**Resumo.** *Este trabalho analisa o uso de uma técnica de redução de dimensionalidade, o Autoencoder, em problemas de classificação de imagens. Para tanto, foram realizados experimentos visando comparar o desempenho do aprendizado quando é utilizada uma rede neural com camadas convolucionais para classificar as imagens originais e outra rede com a mesma arquitetura para classificar as representações geradas pelo Autoencoder. Os resultados são comparados quanto à perda de acurácia e à diferença do tempo de treino.*

## 1. Introdução

A quantidade e a complexidade dos dados usados nas mais diversas áreas têm aumentado consideravelmente. É possível dizer que os dados passaram a possuir uma riqueza maior de informações, como imagens com resoluções cada vez melhores, ou séries temporais com muitos valores em um mesmo intervalo.

Entretanto, manipular um volume denso de dados é uma tarefa complexa, e pode inviabilizar a aplicação. Além disso, nem toda informação em um dado é relevante, ou até mesmo confiável, o que pode ocasionar inconsistências.

Com o intuito de redimensionar os dados para diminuí-los, ou para descartar informações irrelevantes, é concebido o modelo de um Autoencoder. Baseado em uma rede neural, esse modelo recebe os dados como entrada e ao final da sua arquitetura tenta reconstruí-los. Porém, da camada inicial até uma camada intermediária, a dimensão do dado diminui, e por fim volta ao tamanho original. A primeira metade do Autoencoder é a parte relevante do modelo, o encoder, responsável por diminuir a dimensão dos dados, que será analisado neste trabalho.

Para podermos saber a eficiência do Autoencoder, foram feitos dois classificadores de imagens, um para imagens em seu tamanho natural, e outro para as representações das imagens com a dimensionalidade reduzida pela Autoencoder.

O funcionamento e a configuração do Autoencoder, o conjunto de dados utilizado, a descrição dos experimentos e os resultados serão abordados nas próximas seções.

## 2. Autoencoder

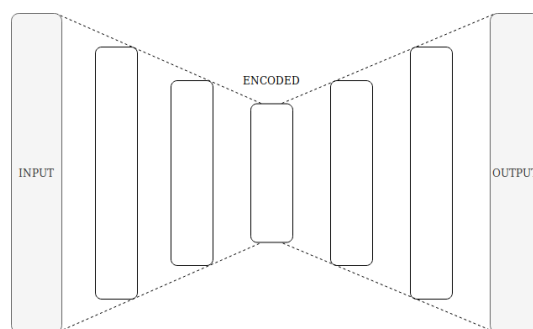
O Autoencoder é uma rede neural que tem por objetivo criar uma representação enxuta da entrada, depois reconstruí-la em seu tamanho original.

Neste trabalho, para melhor compreensão e visualização, foram utilizadas imagens como dados, por isso foram usadas camadas de convolução. Camadas de convolução

são utilizadas de forma que cada neurônio de uma certa camada corresponda a um filtro que será aplicado a um determinado pedaço da imagem. A cada conexão de um conjunto de pixels com um neurônio é atribuído um peso, e a combinação desse peso com os valores de entrada do neurônio produz uma saída que é passada para a camada seguinte[Ana Caroline Gomes Vargas 2016].

A entrada do Autoencoder é uma imagem de 128x128 pixels no padrão RGB. Possui três camadas de convolução, entre duas camadas de maxpooling, que vão diminuindo a dimensão da imagem até 32x32 pixels, e depois duas camadas de convolução, com duas camadas de upsampling, terminando com uma camada de convolução de dimensões 128x128 pixels. Possui um fator de compressão igual a 16.

Entretanto, o mecanismo relevante no Autoencoder é a primeira metade da sua arquitetura, que reduz a dimensão do dado, chamado de encoder. Fisicamente possui as mesmas camadas da primeira metade do Autoencoder, e sua saída é a camada com dimensão 32x32 pixels. Quando o Autoencoder é treinado, o encoder automaticamente também é treinado. No experimento com imagens de formas geométricas, o Autoencoder foi treinado por cinco épocas e com o outro conjunto de dados foi treinado por vinte épocas.



**Figura 1. Arquitetura do Autoencoder.**

### 3. Conjuntos de Dados

Foram utilizados dois conjuntos de dados nos experimentos. O primeiro é o Natural Images<sup>1</sup>, contituído por 6899 imagens de oito classes diferentes: avião, carro, gato, cachorro, flor, fruta, moto e pessoa. O outro conjunto utilizado é o Four Shapes<sup>2</sup>, constituído por 16000 imagens pertencentes a quatro classes distintas: quadrado, estrela, círculo e triângulo.

### 4. Experimentos

Para avaliar a eficiência das representações criadas pelo encoder, foram realizados dois experimentos de classificação de imagens para cada conjunto de dados. Como o intuito é observar o comportamento dos dados, neste trabalho não houve uma preocupação rigorosa com a qualidade dos classificadores, embora seja um diferencial. Seguindo essa premissa, os dois classificadores criados possuem uma arquitetura muito similar, exceto pelo tamanho dos dados que recebem como entrada.

<sup>1</sup><https://www.kaggle.com/prasunroy/natural-images>

<sup>2</sup><https://www.kaggle.com/smeschke/four-shapes>



**Figura 2. Arquitetura dos classificadores.**

#### 4.1. Classificação das imagens originais

Primordialmente, foi criado um classificador para as imagens originais com dimensões 128x128 pixels. Esse classificador possui três camadas convolucionais, com duas camadas de maxpooling entre elas, duas camadas densas e mais uma densa ao final. Nessa última camada, o número de neurônios depende do número de classes do conjunto de dados utilizado, ou seja, oito para o Natural Images e quatro para o outro. Foi treinado por doze épocas com um batch size de tamanho sessenta e quatro.

#### 4.2. Classificação das representações geradas pelo encoder

Para uma comparação justa entre a classificação dos dois tipos de imagem, a arquitetura dos classificadores foi mantida, exceto pela dimensão dos dados. O classificador de representações também possui três camadas convolucionais, duas camadas de maxpooling, duas camadas densas e a camada final com a mesma configuração que o classificador anterior. Nesse cenário, o classificador foi treinado por vinte épocas com um batch size de tamanho sessenta e quatro.

### 5. Resultados

#### 5.1. Métricas

Ao utilizar as imagens em seu tamanho original como entrada, obteve-se uma acurácia ligeiramente maior do que quando foram utilizadas as imagens com dimensão reduzida para ambos os conjuntos de dados. Para o segundo conjunto de dados, devido à sua menor complexidade - possui somente quatro classes - houve uma perda de acurácia de aproximadamente 0.026. Já para o primeiro conjunto de dados, devido à sua maior complexidade - oito classes - houve uma perda maior na acurácia, de aproximadamente 0.073.

**Tabela 1. Acurácia obtida com cada classificador**

Classificador	D1	D2
Original	<b>0.8310</b>	<b>0.9987</b>
Encoded	0.7584	0.9732
Perda de acurácia	<b>0.0726</b>	<b>0.0255</b>

**Tabela 2. Tempo de treino de cada classificador e Autoencoder**

Classificador	D1	D2
Original	24m46s	20m25s
Encoded	<b>6m3s</b>	<b>1m50s</b>
Autoencoder	1h18m	19m11s
Diferença de tempo do treino	<b>18m43s</b>	<b>18m35s</b>

As diferenças de resultados quanto ao tempo do treinamento foram mais significativas. Para o primeiro conjunto de dados, o uso da entrada com menor dimensão resultou

em um treinamento aproximadamente quatro vezes mais rápido, enquanto que para o segundo conjunto de dados, o treinamento foi quase dez vezes mais rápido.

Apesar dessa considerável diferença nos tempos de treinamento dos classificadores, é importante ressaltar que, ao aplicar o autoencoder, também é necessário levar em consideração seu tempo de treinamento. Para o primeiro conjunto de dados, levou-se uma hora e 18 minutos para o treinamento, enquanto que para o segundo, levou-se 19 minutos e 11 segundos.

## **5.2. Interpretabilidade dos dados condensados**

Foi possível perceber que os dados condensados possuem uma similaridade com seus dados originais respectivos, ou seja, as representações não são abstrações quaisquer. Apesar das redes neurais não serem simbólicas, o que buscamos no Autoencoder é a interpretabilidade da representação e não necessariamente do processo.

## **6. Conclusão**

A partir dos experimentos realizados, conclui-se que a utilização ou não de autoencoders para redução da dimensionalidade dos dados antes da tarefa de classificação depende do problema que se quer resolver. Algumas das vantagens de seu uso são o gasto menor de memória, apresentar uma velocidade maior no treinamento dos dados para classificação e diminuir a complexidade do classificador, uma vez que os dados de entrada são apenas uma representação condensada dos dados originais.

## **Referências**

- Ana Caroline Gomes Vargas, A. P. e. C. N. V. (2016). Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. *Proceedings of the XXIX Conference on Graphics, Patterns and Images*, pages 1–4.
- Chollet, F. (2016). Building autoencoders in keras.
- Sharma, A. (2018). Implementing autoencoders in keras: Tutorial.