

Classificação de flores utilizando Redes Neurais Convolucionais

Walter H. Silva
walter.silva@ufv.br

¹Universidade Federal de Viçosa - Campus Rio Paranaíba
Minas Gerais, BR

Abstract. *This work aims the implementation and analysis of Deep Learning models for the classification of flowers of 102 different species. Classification is a type of learning, which aims to better approximate functions that represent the linear or non-linear separation of data. In this context, the classification of flowers is interesting for several areas such as biology, botany, medicine, etc. Convolutional Neural Networks are a category of algorithms in the Deep Learning field, which proved to be excellent algorithms as opposed to the classical Machine Learning methods, generating an accuracy of 97% in flower classification.*

Resumo. *Este trabalho visa a implementação e análise de modelos de Aprendizagem Profunda para a classificação de flores de 102 espécies diferentes. Classificação é um tipo de aprendizagem, que visa melhor aproximar funções que representem a separação linear ou não linear de dados. Neste contexto, a classificação de flores se mostra interessante para diversas áreas como a biologia, botânica, medicina, etc. Redes Neurais Convolucionais são uma categoria de algoritmos do campo de Aprendizagem Profunda, que se mostraram excelentes algoritmos em contraposição aos métodos clássicos de Aprendizagem de Máquina, gerando uma acurácia de 97% na classificação de flores.*

1. Introdução

Iris [Dheeru and Karra Taniskidou 2017] é um conjunto de dados de flores lançado em 1936, que possui 4 atributos referentes as características de 3 espécies flores. Em razão de sua simplicidade o conjunto é utilizado como exemplo para o ensino em áreas como o Aprendizado de Máquina, Aprendizagem Profunda, Mineração de Dados, etc. O conjunto de dados de flores de 102 espécies [Nilsback and Zisserman 2008] tem o intuito de desafiar os modelos de Aprendizagem Profunda contemporâneos, uma vez que o mesmo possui 102 espécies e sua complexidade se eleva consideravelmente se comparado ao conjunto Iris.

O desafio em questão é proposto como projeto final do curso online da Udacity e do curso presencial de Aprendizagem Profunda da Universidade Federal de Viçosa - Campus Rio Paranaíba. Diferentes arquiteturas de Redes Neurais Convolucionais, assim como seus hiperparâmetros serão abordados a fim de obter uma melhor acurácia na separação das espécies de flores e por conseguinte auxiliar na tarefa de classificação realizada por humanos, que se encontra a merce do viés ou erro humano. A utilização de Redes Neurais Convolucionais se faz presente em razão da excelente capacidade das mesmas em lidar com a extração de características, onde a relação entre as variáveis é complexa.

2. Configuração dos Experimentos

2.1. Conjunto de dados

2.1.1. Treinamento

O conjunto de dados de flores de 102 espécies [Nilsback and Zisserman 2008] foi utilizado como treinamento. Por se tratar de um problema de classificação, as classes atribuídas ao conjunto de dados em questão foram utilizadas. O conjunto de treinamento se consiste em 102 espécies diferentes de flores, cada uma contendo de 40 a 248 imagens RGB de largura 400-700 pixels e altura 500 pixels, total de 6552 imagens.

As imagens em questão foram redimensionadas, transformadas para tensores e normalizadas, além disso o aumento de dados foi aplicado, gerando 5 novos conjuntos de dados:

1. Conjunto 2: imagens rotacionadas entre -30° e 30°
2. Conjunto 3: imagens transformadas para brilho 0-100%, hue 50% e saturação 50%
3. Conjunto 4: imagens recortadas randomicamente, largura 224 pixels e altura 224 pixels
4. Conjunto 5: imagens invertidas horizontalmente
5. Conjunto 6: imagens invertidas verticalmente

Após o aumento de dados a quantidade total de imagens é de 39,312 mil imagens. A figura 1 ilustra parte do conjunto de treinamento.

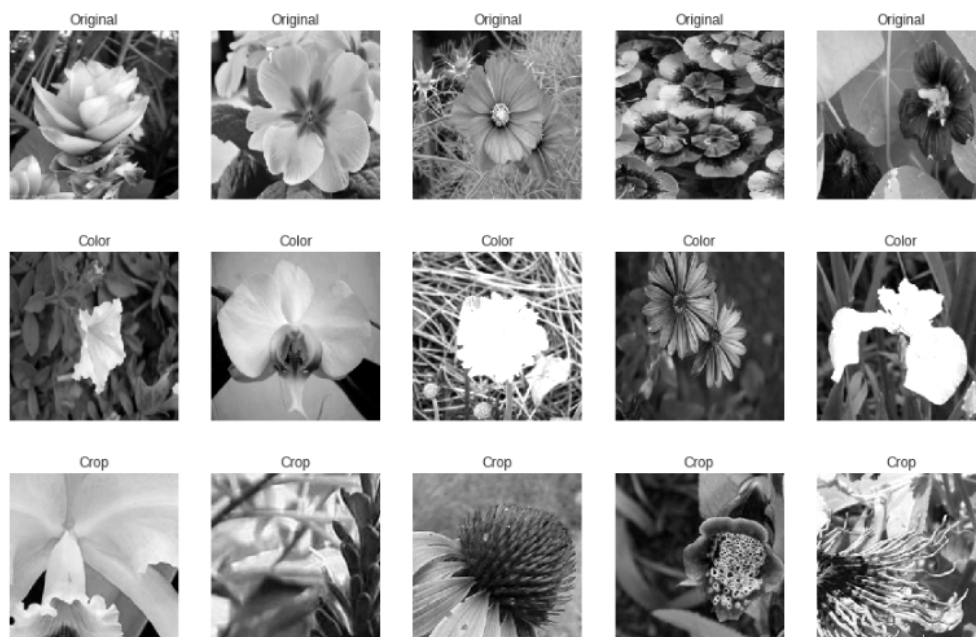


Figure 1. Treino

2.1.2. Validação

O conjunto de validação se consiste de 818 imagens, diferentes das imagens de treinamento não geradas artificialmente, representando 102 espécies diferentes de flores. Imagens RGB de largura 400-800 pixels e 500-800 pixels de altura. As imagens de validação foram redimensionadas, transformadas para tensores e normalizadas. A figura 2 ilustra parte do conjunto de validação.

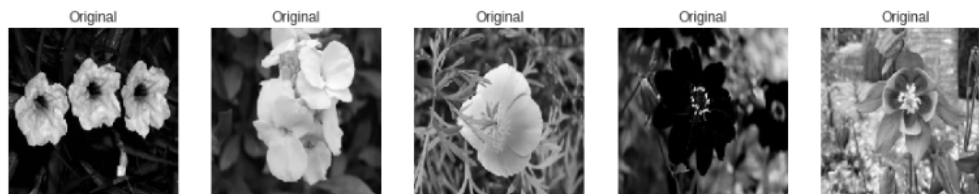


Figure 2. Validação

2.1.3. Testes

As imagens de testes são 20% das imagens de validação, contudo neste trabalho as mesmas não foram utilizadas, ou seja, o conjunto de validação não foi alterado.

2.2. Ferramentas

PyTorch é um framework criado por engenheiros do Facebook com o intuito de facilitar a criação de modelos de alto nível utilizando arquiteturas consideradas o estado da arte, além de ser modularizado e extremamente rápido, possuindo uma API escrita em C++ e ONNX, utilizada para a exportação dos modelos. Alternativas como o framework Keras e Sonet foram consideradas, contudo PyTorch se mostrou uma melhor opção em ambas as características mencionadas do mesmo.

Os experimentos foram executados através de uma GPU Tesla K80 e uma GTX Mx 150, utilizando-se da tecnologia CUDA.

2.3. Arquiteturas

2.3.1. ResNet 18

ResNet 18 [He et al. 2016] é uma arquitetura de Rede Neural Convolutacional de 18 camadas, que evita o desaparecimento do gradiente, isso é realizado através da adição da entrada da camada anterior a saída da camada atual, a figura 3 ilustra o processo descrito. A arquitetura em questão foi configurada para utilizar o aprendizado do treino do conjunto de dados Image Net, assim como as duas primeiras camadas da Rede Neural foram congeladas.

2.3.2. VGG 11

VGG 11 [Simonyan and Zisserman 2014] é uma arquitetura de Rede Neural Convolutacional de 11 camadas, que diminui o custo computacional através da definição de

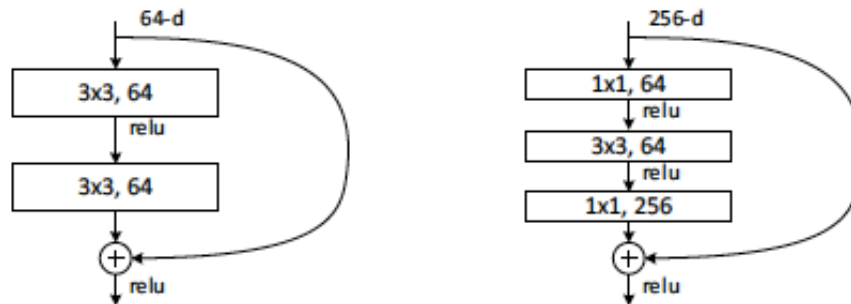


Figure 3. ResNet 18 - Arquitetura

filtros convolucionais com tamanho específico durante toda a Rede Neural, a figura 4 ilustra a arquitetura descrita. A arquitetura em questão foi configurada para utilizar o aprendizado do treino do conjunto de dados Image Net, assim como o primeiro bloco de convolução foi congelado.

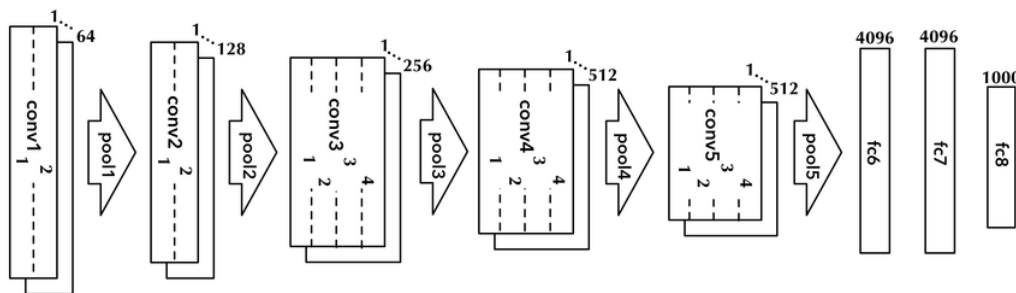


Figure 4. VGG 11 - Arquitetura

3. Resultados e discussões

3.1. ResNet 18

A Rede Neural Convolucional ResNet 18 foi inicialmente treinada com batches de tamanho 124, otimizador SGD com taxa de aprendizado de 0.001 e todos os conjuntos de dados, obtendo 87% de acurácia, contudo em razão dos altos valores nos hiper parâmetros mencionados, a rede não performa como esperado. Ao reduzir o tamanho dos batches para 12, otimizador ADAM, que se adapta ao longo da propagação da rede, com taxa de aprendizado de 0.0001 e todos os conjuntos de dados, a taxa de acurácia foi de 97%, assim como a figura 5 ilustra.

```
[22] 1 print('Model accuracy: {}'.format(round(100. * np.sum(resnet18_class_correct) / np.sum(resnet18_class_total), 2)))
```

Model accuracy: 97.0

Figure 5. ResNet 18 - Acurácia

Decrementar os hiperparâmetros mencionados causa uma taxa de aprendizado lenta, contudo concisa, que evita o desaparecimento do gradiente, ou seja, o gradiente não para na local mínima.

O gráfico da perda do treinamento ilustra picos, que são ocasionados em razão da entrada de um novo conjunto de dados na fase de treinamento, isso se trata de uma hipótese, que pode ser confirmada randomizando todos os conjuntos de treinamento.

Nenhuma semente foi definida, contudo o modelo foi treinado múltiplas vezes para garantir o resultado de 97% de acurácia obtido, treinado com múltiplas combinações de hiperparâmetros e conjuntos e ordem dos conjuntos de treinamento, assim como o treinamento foi salvo e encontra-se disponível no repositório do Github [Sphinxs 2019].

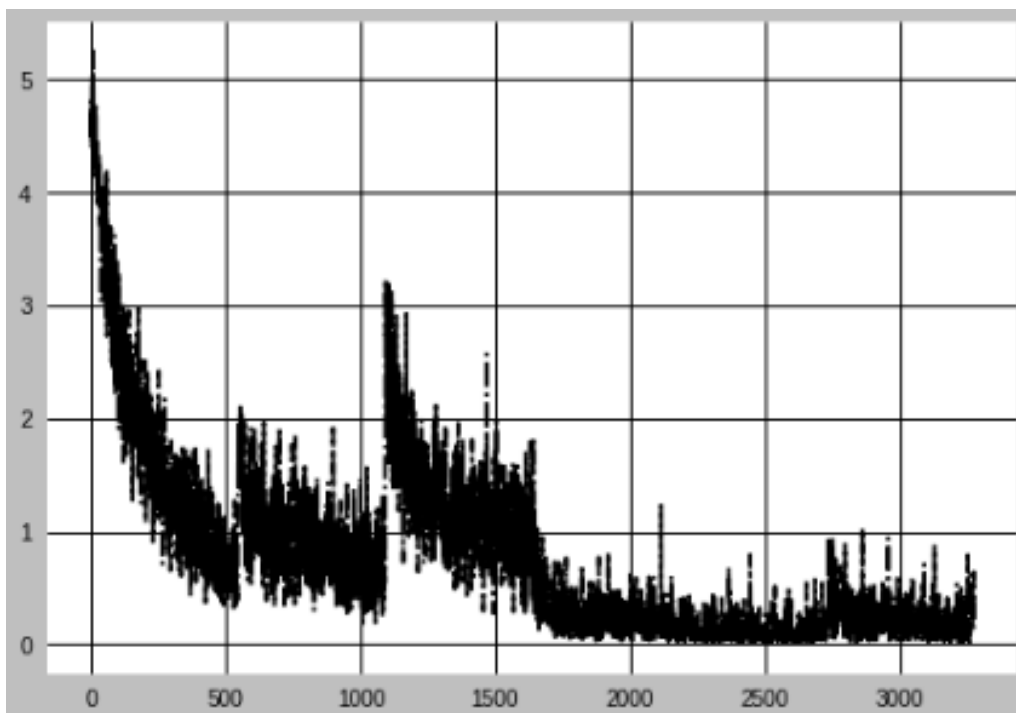


Figure 6. ResNet 18 - Perda

3.2. VGG 11

A partir do aprendizado de como treinar um modelo, obtido no treinamento da rede ResNet 18, o treinamento da rede VGG 11 se deu inicialmente através de batches de tamanho 12, otimizador ADAM com taxa de aprendizado de 0.0001 e todos os conjuntos de dados, obtendo 87% de acurácia, contudo neste caso apenas o tamanho de batch se fez um bom parâmetro, uma vez que tal rede requer um poder computacional vezes maior do que a rede ResNet 18. Para a melhoria do modelo foi aplicado o otimizador SGD com taxa de aprendizado de 0.001 e batches de tamanho 12, obtendo 92% de acurácia, assim como a figura 7 ilustra.

```
[24] 1 # Plot accuracy
      2 vgg11_class_correct, vgg11_class_total, vgg11_sample = accuracy(vgg11, validation, classes)
      3
      4 print('Model accuracy: {}%'.format(round(100. * np.sum(vgg11_class_correct) / np.sum(vgg11_class_total), 2)))
```

Model accuracy: 92.0

Figure 7. VGG 11 - Acurácia

Uma taxa de aprendizado maior demonstra-se interessante para a arquitetura VGG 11, por mais que a mesma não possa ser intuitiva, uma vez que pode ocasionar o desaparecimento do gradiente, ou seja a parada ou loop na local mínima.

A mesma característica referente ao treinamento com múltiplos conjuntos de dados se fazem presente no treinamento da rede VGG 11, como é ilustrado na figura 8.

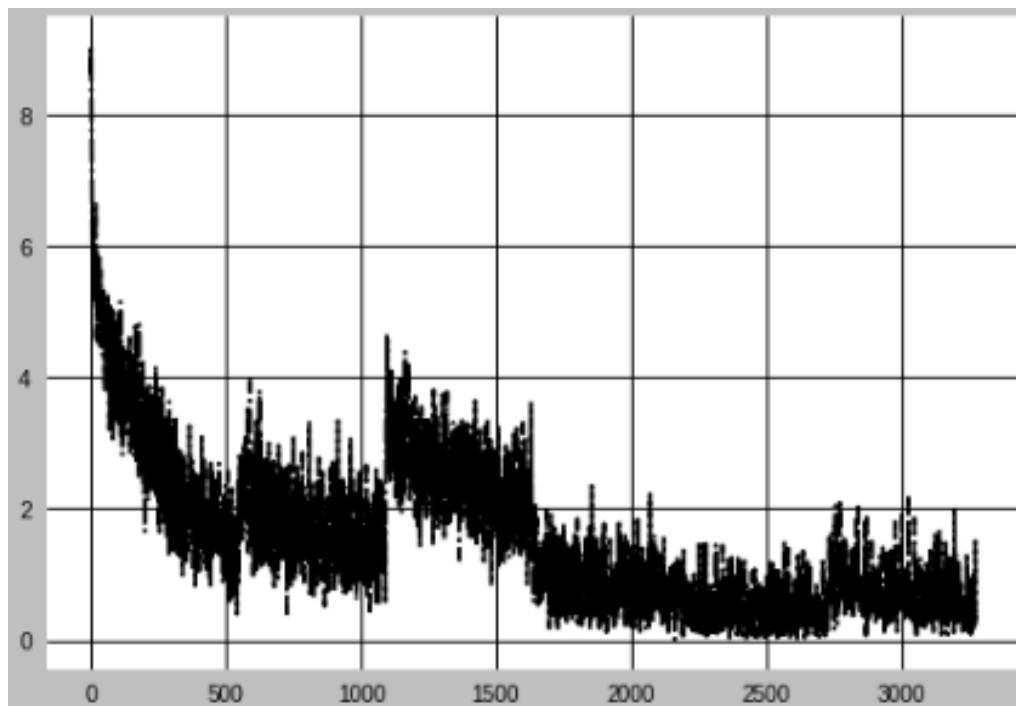


Figure 8. VGG 11 - Perda

4. Conclusão

Conclui-se que o processamento e escolha de arquitetura, assim como a configuração de seus hiper parâmetros, causam um grande impacto no resultado final. Nete trabalho o aumento de dados, uma baixa taxa de aprendizado e batchs pequenos ocasionaram um melhor resultado, contudo tais aspectos também ocasionam um grande custo computacional.

Buscar um abordagem de meta aprendizado, aprendizagem por reforço, aprendizagem não supervisionada, etc; pode ser estudado em trabalhos futuros, assim como a utilização de K-Fold, Synthetic Data, randomização dos dados de treinamento, etc.

References

- [Dheeru and Karra Taniskidou 2017] Dheeru, D. and Karra Taniskidou, E. (2017). UCI machine learning repository.
- [He et al. 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

- [Nilsback and Zisserman 2008] Nilsback, M.-E. and Zisserman, A. (2008). Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*.
- [Simonyan and Zisserman 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [Sphinxs 2019] Sphinxs (2019). Vida. <https://github.com/sphinxs/vida>.