

Hands On: Classificação de Imagens com Redes Neurais

Neste hands on você vai precisar de:

- Python 2.7 ou 3.0
- Matplotlib
- Pandas
- Numpy

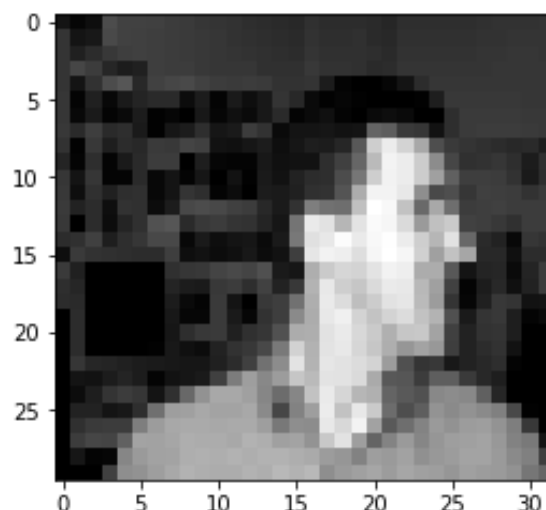
Opcionalmente você pode realizar este hands on a partir do Jupyter ou Spyder.

Iniciando

Você deve clonar o repositório presente em https://github.com/lucasbf/Neural_Nets_Study.git, onde temos todos os recursos que serão necessários para a execução deste hands on.

Dataset

O dataset que será utilizado é uma versão reduzida (imagens 30x32) do dataset criado por Tom Mitchell, o qual já está no repositório (Datasets/Faces/faces_4/). Tal base de dados está organizada em pastas indicada pelo nome de cada um dos voluntários para a criação deste dataset. Cada uma das imagens possui uma foto de um respectivo rosto, e seu respectivo arquivo está nomeado com a indicação do nome da pessoa, orientação para onde sua cabeça está virada, seu humor e como estão seus olhos. Por exemplo, no arquivo `mitchell_left_angry_open_4-ascii.pgm` temos:



Você deve extrair os pixels de todas as imagens e associa-los às suas respectivas classificações com o objetivo de montar uma base de dados no formato csv para seu posterior carregamento no Pandas. Por exemplo, a seguir temos a imagem anterior transformada em instância para a base de dados:

	Id	Col 0	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	...	Col 953	Col 954	Col 955	Col 956	Col 957	Col 958	Col 959	Orientati on	Senti ment	Eyes
29	29	10	25	23	22	20	19	16	15	14	...	114	114	114	113	113	100	26	left	angry	open

Cada uma das imagens recebe um Id (valor autoincrementado), seguido de 960 colunas (30x32), e por fim tem-se as colunas com as três respectivas classificações.

Após a construção da base de dados, carregue a mesma usando o Pandas e a seguir polo algumas imagens para verificar se tudo correu como esperado.

Análises

Faça um gráfico em barra contendo a contagem de cada um dos valores das classes presentes na mesma.

Redes Neurais (Treinamento e Teste)

Para a fase de treinamento da rede neural você poderá usar arquivo `mlp.py` disponível na pasta `Models`. Nessa fase você deverá:

- Criar uma rede neural MLP para cada uma das classificações necessárias, ou seja, 3 redes neurais.
- Avaliar o desempenho de cada MLP a partir da metodologia Train-Test-Split. Construa gráficos com os erros de treinamento e teste.
- Crie e avalie também o desempenho de uma MLP que possa aprender a classificar todos as classes de uma vez.
- Você sabe que uma MLP possui vários parâmetros, tais como quantidade de camadas, quantidade de neurônios por camada, etc., sendo assim faça vários testes para encontrar os melhores parâmetros.
- Extra: você pode utilizar a técnica de validação cruzada, a qual é uma forma de treinamento-teste muito útil, pois superutiliza a base de dados disponível e minimiza o enviesamento de classificação a partir dos dados de teste escolhidos.

Bom divertimento.