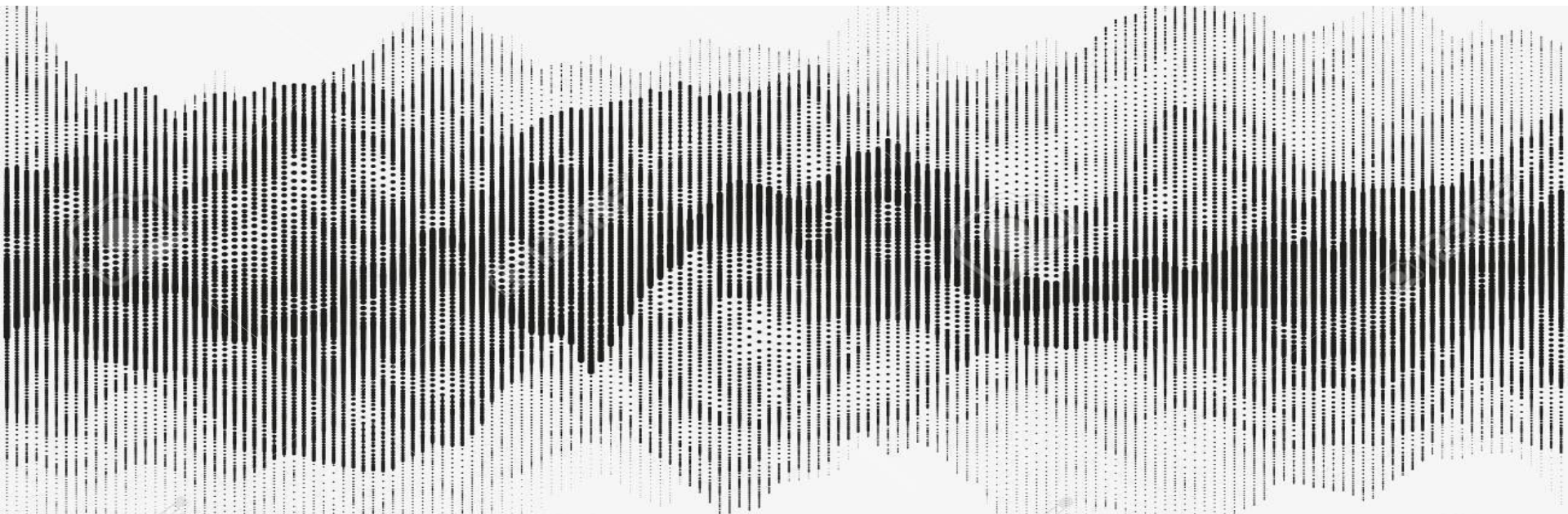


AUDIOCOMANDOS



Beatriz, Diêgo e Diego

1º passo: implementação própria de uma MLP

Implementação simples, não otimizada, seguindo a teoria do livro

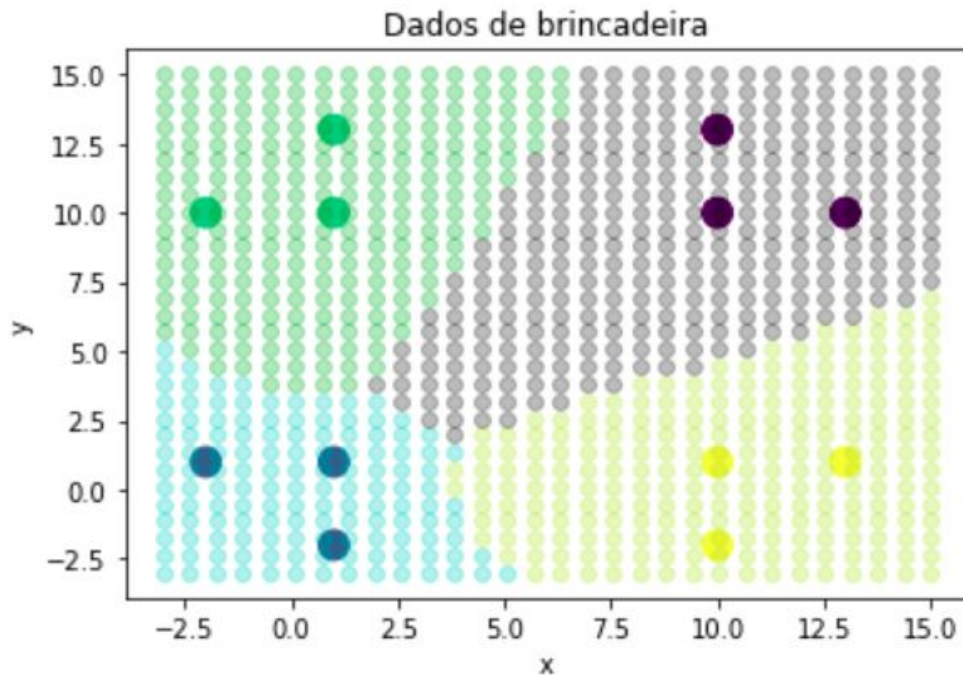
- Definir as dimensões da rede
- Iterar nos dados, fazendo forward e backward pass

```
dimensoes_camadas = [4,6,4]
```

```
erro medio: 1.033392594660492  
erro medio: 0.001067571714454552  
erro medio: 0.0005049324627105386  
erro medio: 0.00032722484381672947  
erro medio: 0.00024095601904046039
```

1º passo: implementação própria de uma MLP

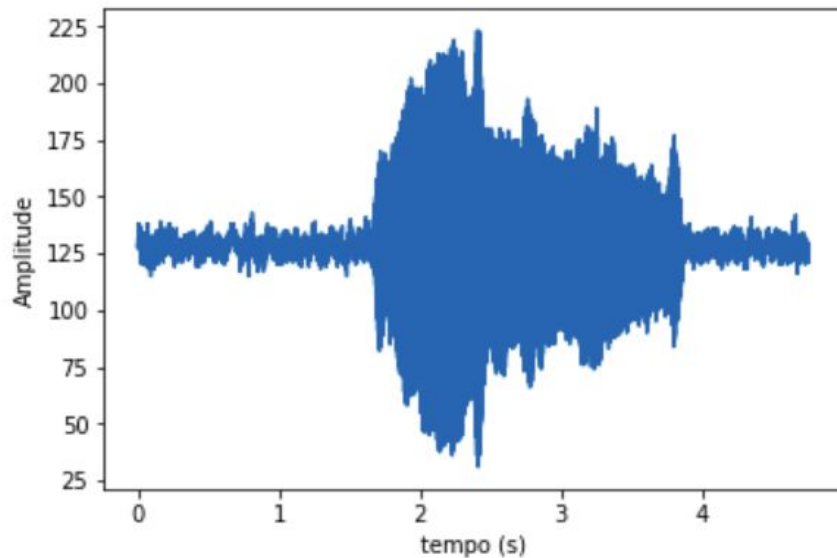
MLP funcionando adequadamente:



2º passo: conhecimento do domínio de aplicação

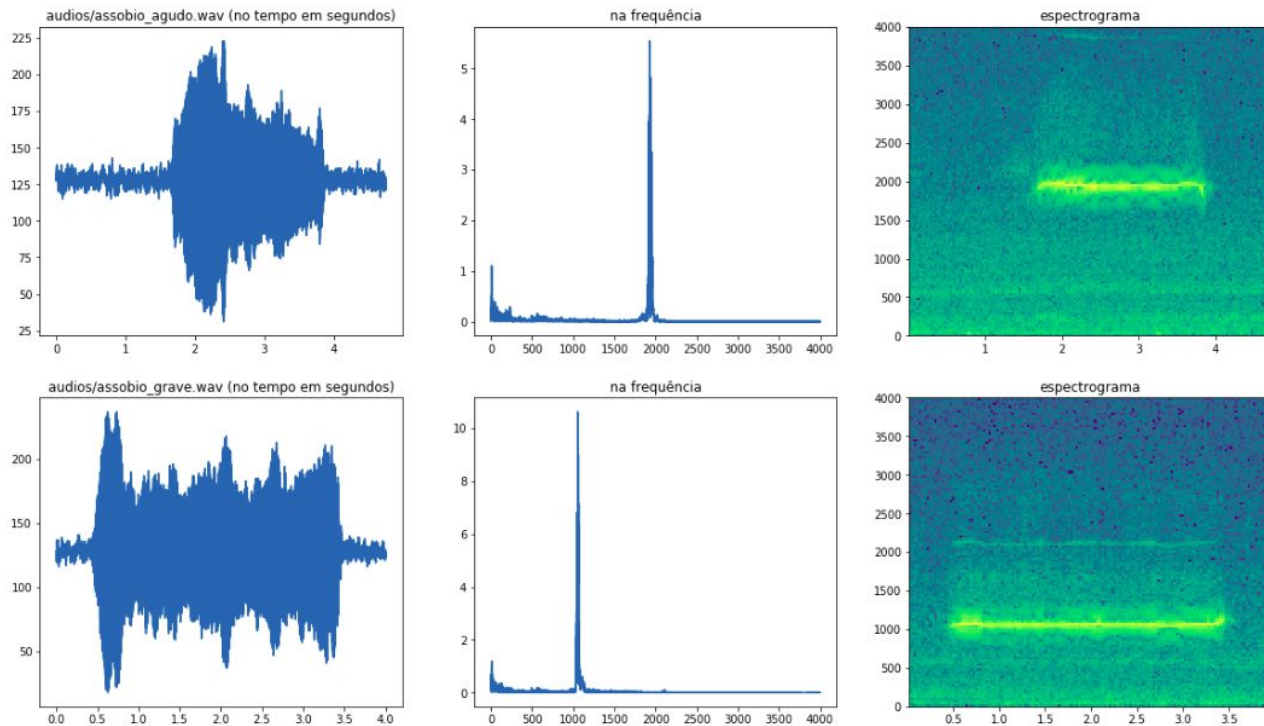
Coletamos alguns áudios e os analisamos para saber como poderíamos processá-los antes de jogar na MLP.

- Gravar um arquivo de áudio
- Acessar o arquivo gravado via código



2º passo: conhecimento do domínio de aplicação

Decidimos fazer (quase) toda a análise no domínio da frequência.



2º passo: conhecimento do domínio de aplicação

Realizamos uma breve revisão bibliográfica em busca de trabalhos parecidos

Automatic Classification of Audio Data*

Carlos H. L. Costa, Jaime D. Valle Jr., Alessandro L. Koerich
Pontifical Catholic University of Paraná
Curitiba, PR, Brazil
alekoe@computer.org

Abstract – *In this paper a novel content-based musical genre classification approach that uses combination of classifiers is proposed. First, musical surface features and beat-related features are extracted from different segments of digi-*

non-parametric classification strategies and have dealt with small databases.

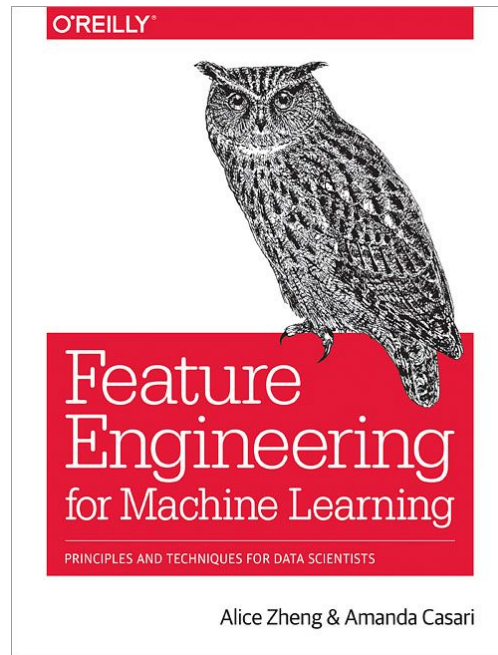
Musical genre is an important description that has been used to classify and characterize digital music and to or-

2º passo: conhecimento do domínio de aplicação

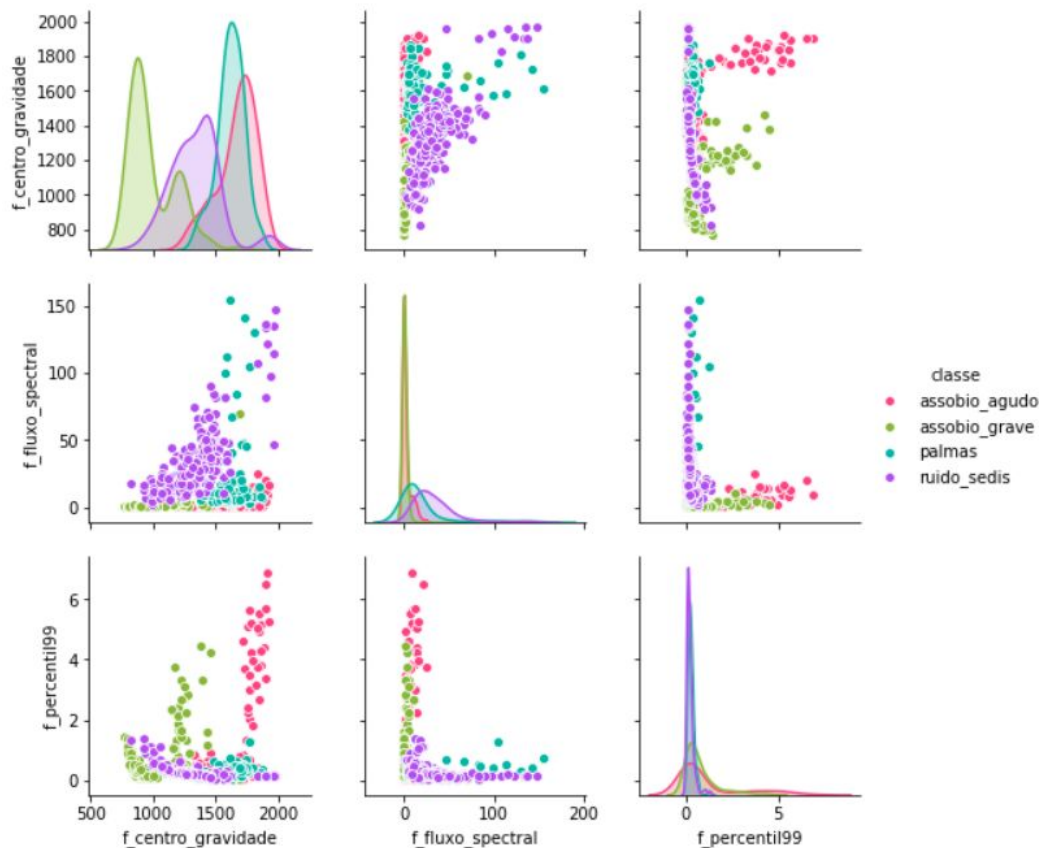
Implementamos uma série de procedimentos de extração de características

Analizamos quais seriam úteis

Coletamos várias amostras (500~600) para
pré-processamento e “estoque”



2º passo: conhecimento do domínio de aplicação



3º passo: ajuste de modelo

Experimentamos algumas arquiteturas com nossa MLP...

Experimentamos transformações nos dados, como padronização...

Uma vez satisfeitos, partimos para testes

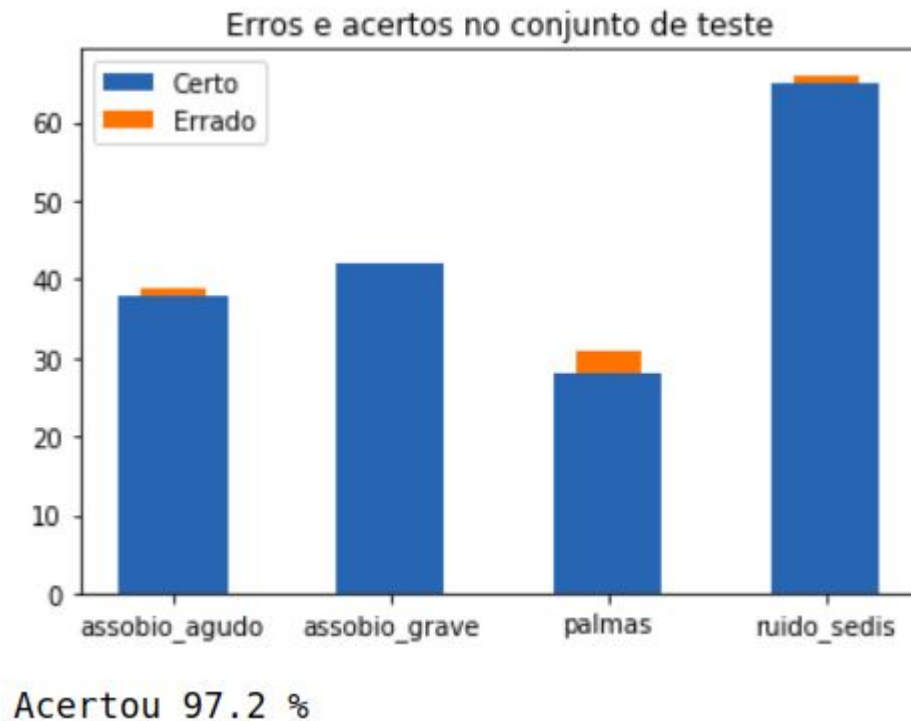
e validação.

```
# Quantos neurônios tem em cada camada (sem contar o bias)  
formato_rede = [4, 10, 4]  
  
# Variação mínima do erro como critério de parada  
delta_erro_minimo = 0.00001  
  
# Número máximo de iterações  
max_iteracoes = 10000
```

4º passo: avaliação

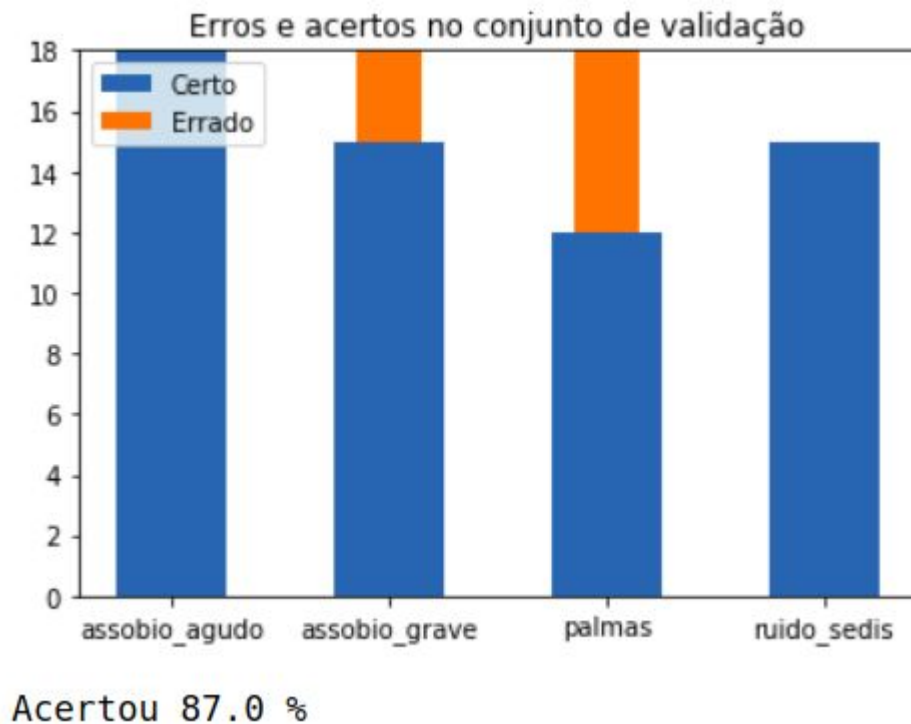
O treino e o teste foram realizados com conjuntos separados.

O modelo aprendeu bem a reconhecer aquilo que lhe foi mostrado



4º passo: avaliação

Na validação, utilizamos um conjunto de dados totalmente novo, que o modelo nunca teve a chance de ver, para verificar se o modelo generalizava.



5º passo: produto final

Portamos o modelo treinado para um programa separado, que acessava a entrada de áudio no microfone do notebook e detectava em tempo real os comandos.

