

NEUROCOMPUTAÇÃO NA CLASSIFICAÇÃO DE FALHAS EM MÁQUINAS ROTATIVAS

PSI5886 – Princípios de Neurocomputação

Prof. Emilio Del Moral Hernandez
Prof. Marcio Lobo Netto

Integrantes do grupo



Amanda Lucatto
Marra



Carlos Eduardo
Motta Tassinari



Gerson Yuri
Cagnani Conte



Murilo Turquiai
Luca Blasio

- Escopo do trabalho e dos ensaios definidos em conjunto
- Alinhamento de embasamento teórico em conjunto
- Cada integrante focou em um tipo de ensaio:
 - Gerson: MLP com dados crus
 - Murilo: MLP com espectros
 - Amanda: Convolucional com dados crus
 - Carlos: Convolucional com espectros
- Porém, houve colaboração de todos os integrantes em todos os ensaios de forma que todos tiveram contato com todos os ensaios
- Comparação de resultados e conclusão em conjunto

Agenda

- Introdução ao Domínio
- Redes Neurais Convolucionais
- Ensaios realizados
 - Plataforma utilizada
 - Dataset
 - Seleção e compensação dos dados
 - Funções de perda
 - MLP com dados crus
 - MLP com espectros de frequência
 - Convolucionais 1D com dados crus
 - Convolucionais 1D com espectros de frequência
- Comparações entre os resultados dos ensaios
- Conclusões
- Outras aplicações

INTRODUÇÃO AO DOMÍNIO

Análise de vibração de máquinas rotativas

Máquinas rotativas

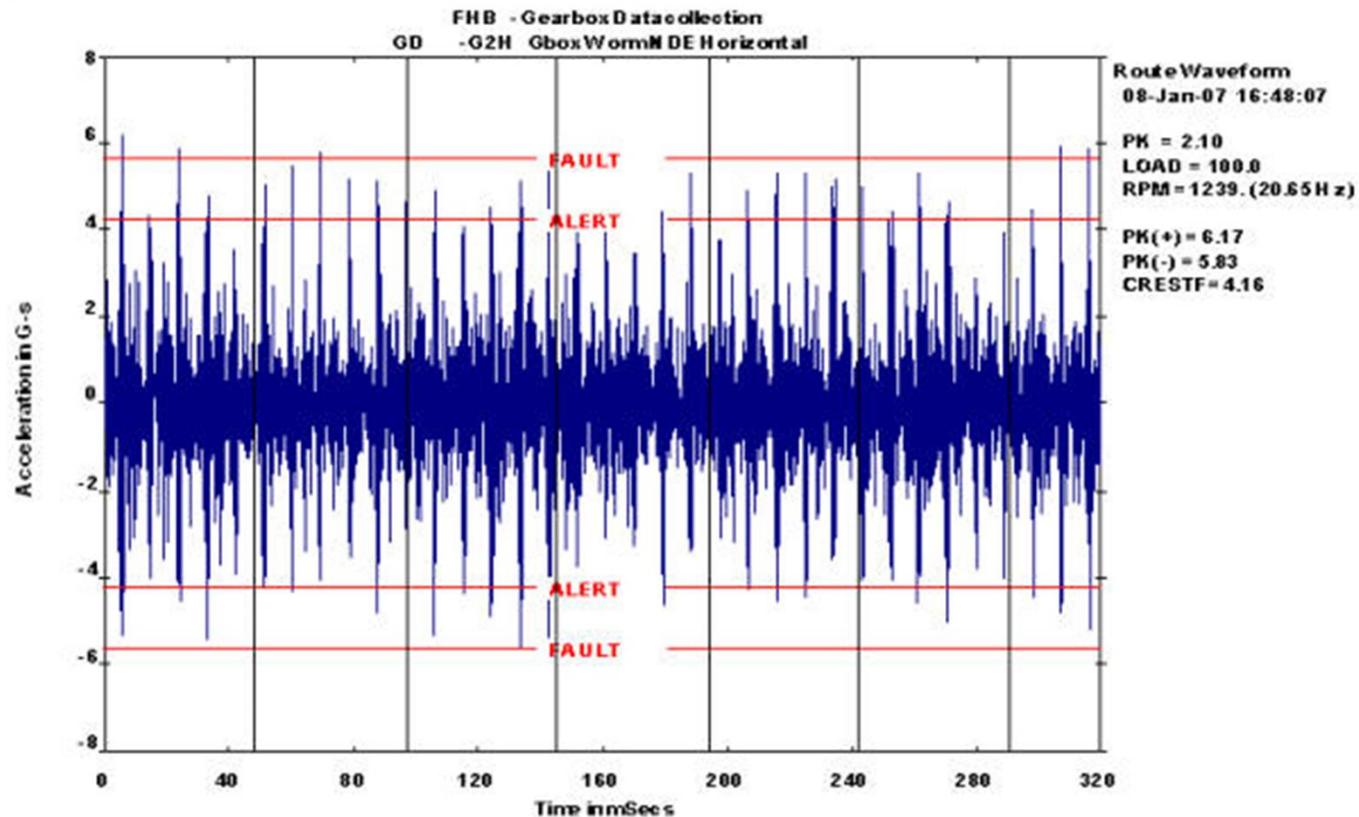
- O que são máquinas rotativas?
 - Geradores de energia
 - Motores
 - Compressores de ar
 - Turbinas termelétricas
 - Turbinas eólicas
 - Etc.



Fonte: Acervo pessoal

Monitoração da condição

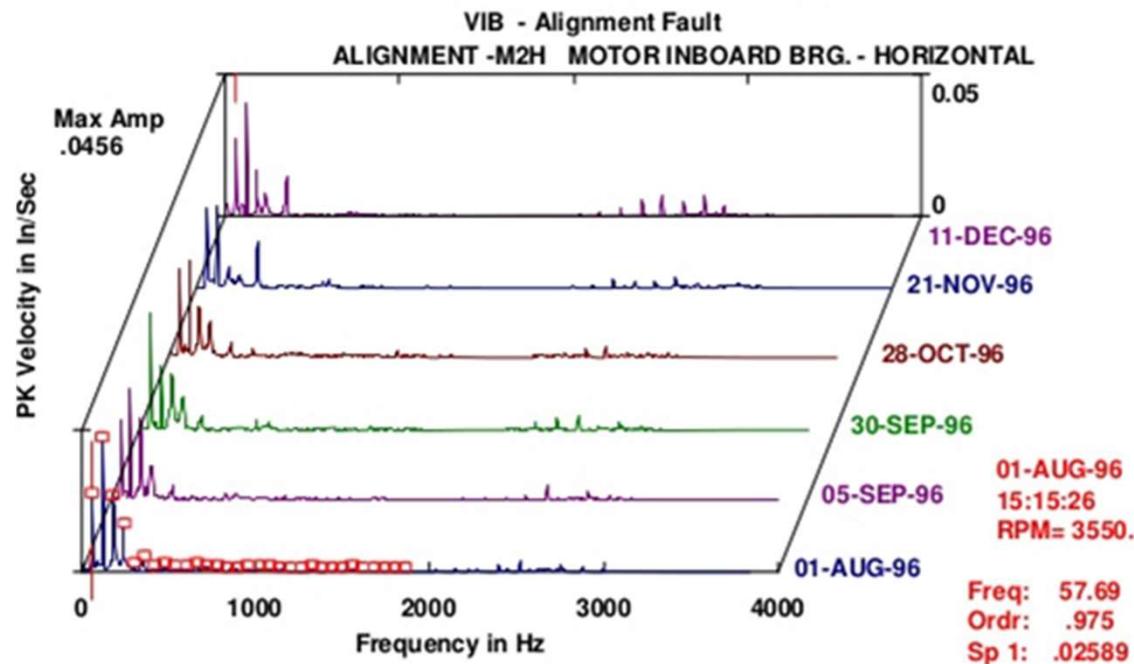
- Todos equipamentos rotativos de grande porte monitoram vibração



Fonte: <https://www.designworldonline.com/conformal-coating-for-power-supplies/>

Análise do sinal de vibração

- Espectro de frequência
- Comparação visual periódica por um especialista
- Técnicas de análise e conhecimento da dinâmica do equipamento

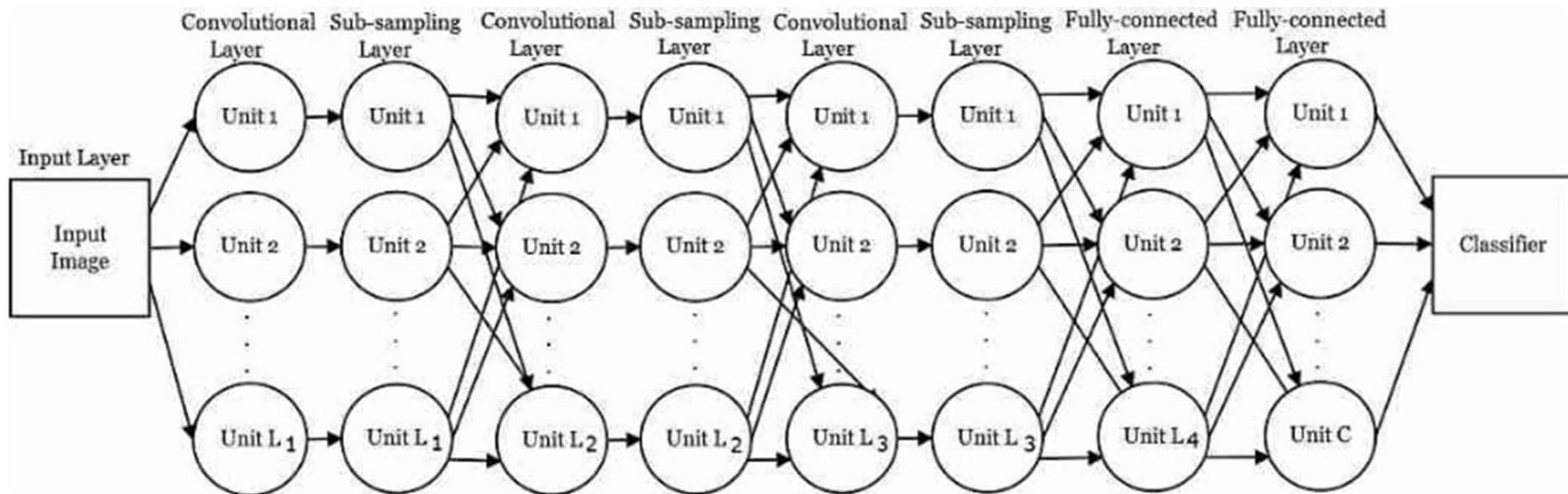


Fonte: <https://www.slideshare.net/makarandnikume/rt-2008-principles-and-practices-of-vibrational-analysis-keeper>

REDES CONVOLUCIONAIS

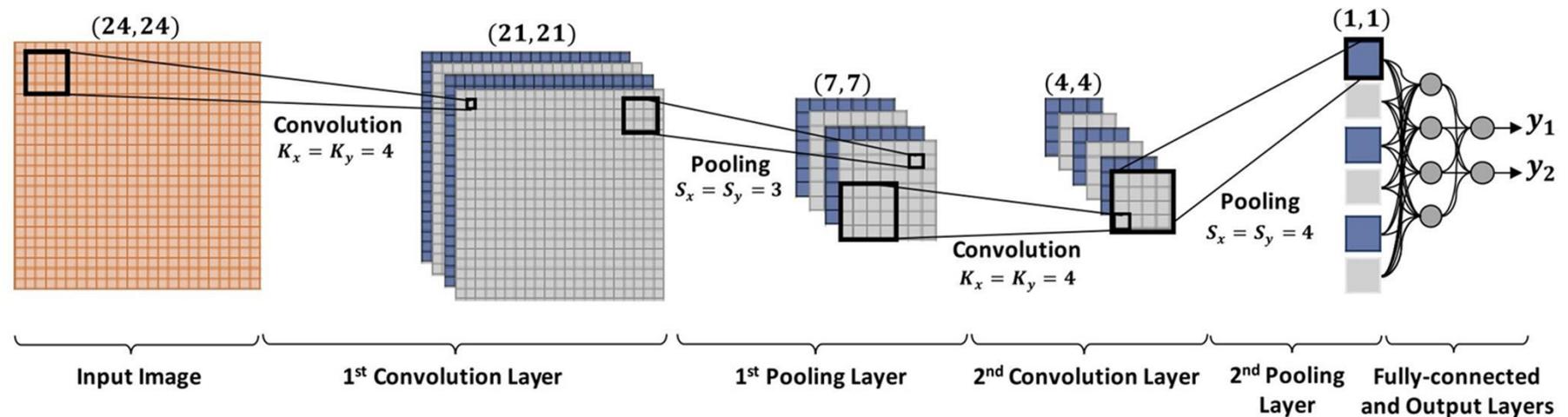
Redes convolucionais

- Redes neurais *feed-forward* com camadas convolucionais e camadas de subamostragem que alternam entre si.



Fonte: (FERREIRA, 2019, p. 46)

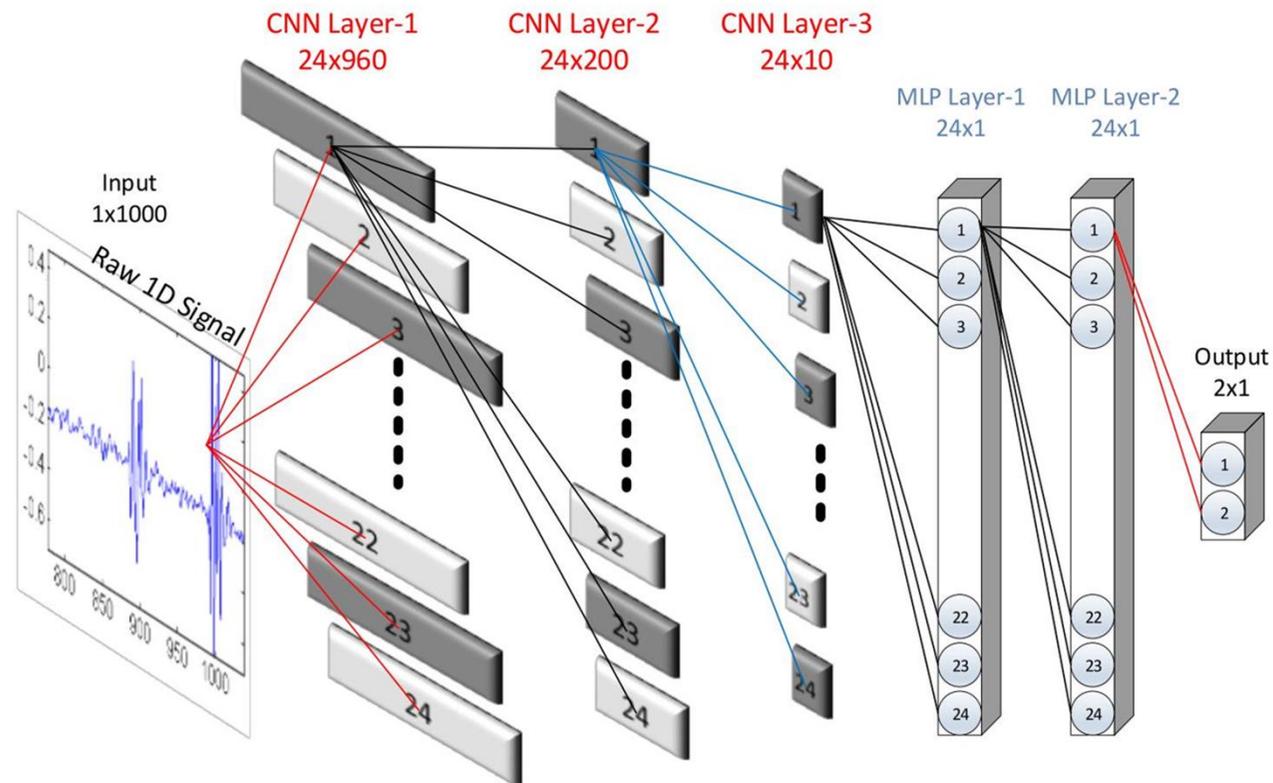
Redes convolucionais



Fonte: (KIRANYAZ et al., 2019)

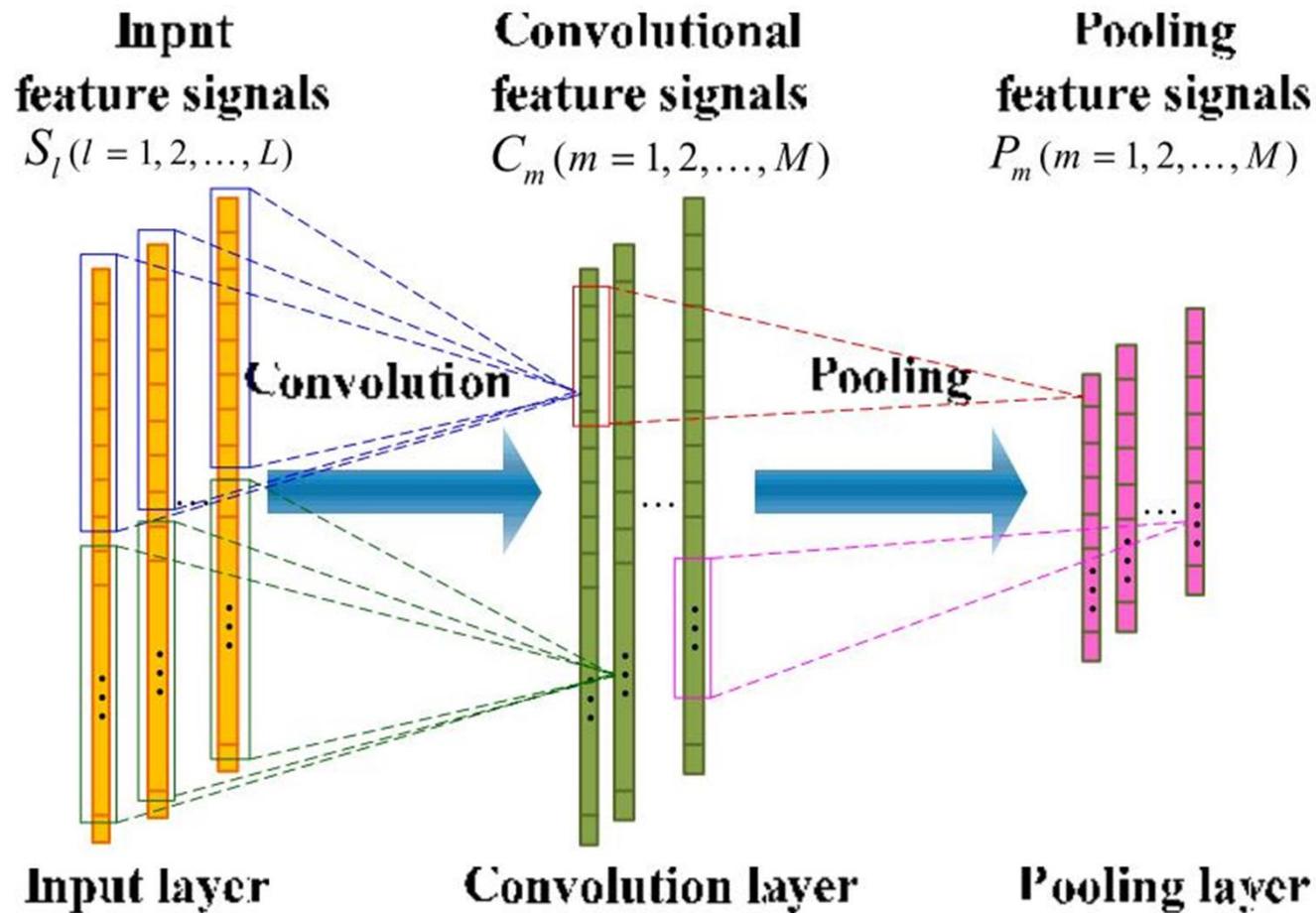
- Criadas especificamente para sinais 2D
- Imagens e vídeos

Redes convolucionais 1D



Fonte: (KIRANYAZ et al., 2019)

Redes convolucionais 1D



Fonte: (YANG et al., 2019)

Redes convolucionais 1D

“A pooling function replaces the output of the net at a certain location with a summary statistic of the nearby outputs”.

“Convolution is equivalent to converting both the input and the kernel to the frequency domain using a Fourier transform, performing point-wise multiplication of the two signals, and converting back to the time domain using an inverse Fourier transform. For some problem sizes, this can be faster than the naive implementation of discrete convolution.”

Fonte: (GOODFELLOW, BENGIO, COURVILLE, 2017)

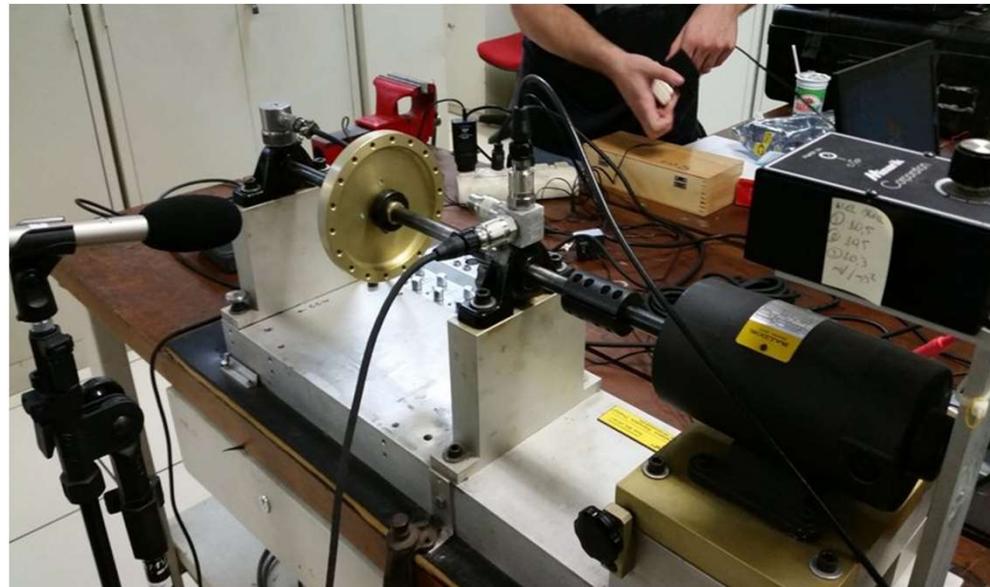
ENSAIOS REALIZADOS

Plataforma

- Combinação entre Google Colab e Jupyter Notebook
 - Limitações de espaço no Google Drive
 - Limitações de processamento de computador pessoal
- Linguagem de programação: Python
- Biblioteca: Keras

Dataset

- MaFaulDa (Machinery Fault Database – SMT – UFRJ)
- 1.951 séries multivariadas simuladas mecanicamente
- 10 condições diferentes
- Baseline (sem defeito), desbalanceamento, desalinhamento e falhas de rolamento.



Fonte: (MARINS et al., 2018)

Dados utilizados

1.4. Sequences
Each sequence was generated at a 50 kHz sampling rate during 5 s, totaling 250.000 samples.

Normal	49
Horizontal misalignment	197
Vertical misalignment	301
Imbalance	333
Underhang bearing	
Cage fault	188
Outer race	184
Ball fault	186
Overhang bearing	
Cage fault	188
Outer race	188
Ball fault	137
Total	1951

Fonte: http://www02.smt.ufrj.br/~offshore/mfs/page_01.html

- Cada arquivo possui 8 colunas:
- COLUNA 1: SINAL DO TACOMETRO (permite estimar a frequência de rotação da máquina)
- COLUNAS 2-4: SINAIS DOS ACELERÔMETRO DO MANCAL INTERNO (nas direções, axial, radial e tangencial)
- COLUNAS 5-7: SINAIS DO ACELERÔMETRO TRIAXIAL DO MANCAL EXTERNO (nas direções axial, radial e tangencial)
- COLUNA 8: SINAL CORRESPONDENTE AO MICROFONE
- **Utilizamos apenas a coluna 2 de todos os arquivos**
- **Total de 683 objetos**

Compensação dos dados

- Dados desbalanceados
- Compensação inserindo pesos nas classes durante o treinamento:

Classe	Objetos	Peso MLP	Peso Convol
Normal	49	5	4
Imbalance	333	0,5	1
Vertical misalignment	301	0,5	1

Fonte: Aatoria própria

Funções de perdas

- Cross entropy

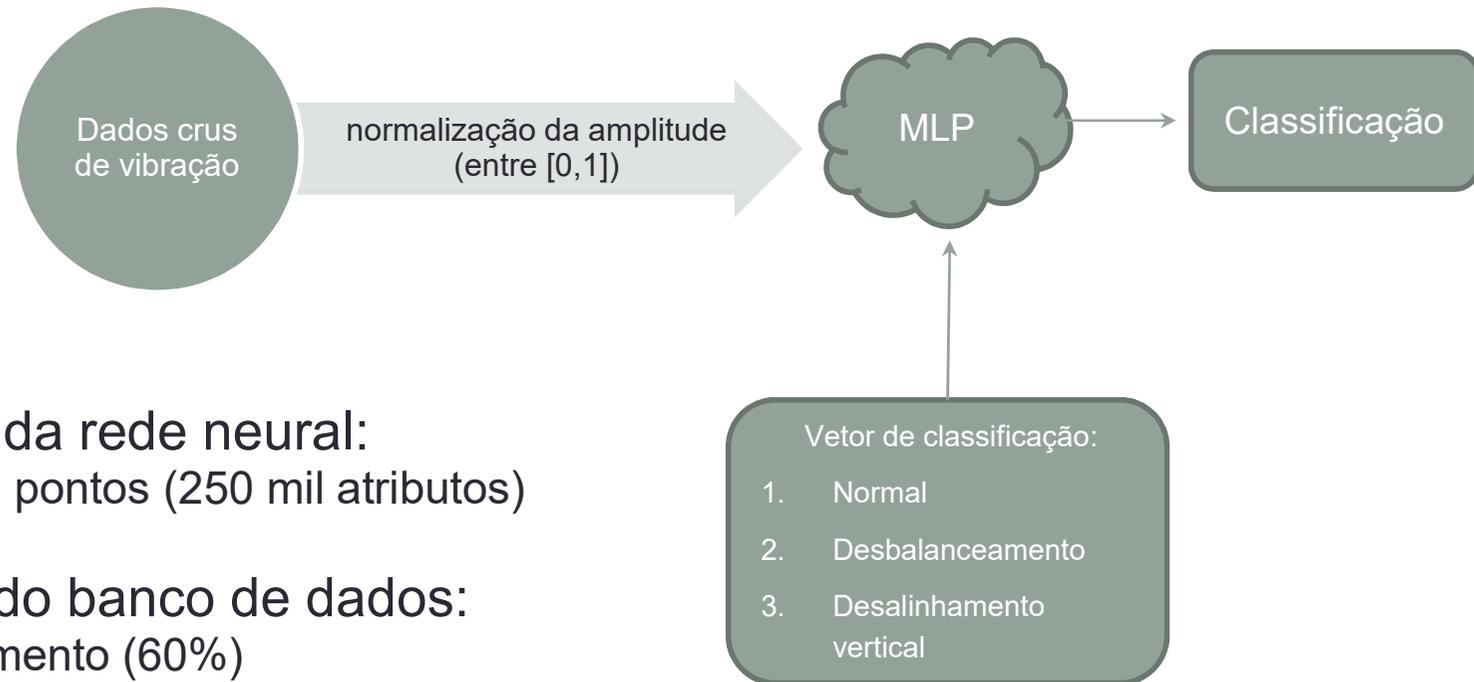
$$H(P, Q) = - \sum_x P(x) \log Q(x)$$

- Kullback Leibler divergence

$$KL(P|Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

$$D_{KL}(A \parallel B) = \sum_i p_A(v_i) \log p_A(v_i) - p_A(v_i) \log p_B(v_i),$$

MLP com dados crus

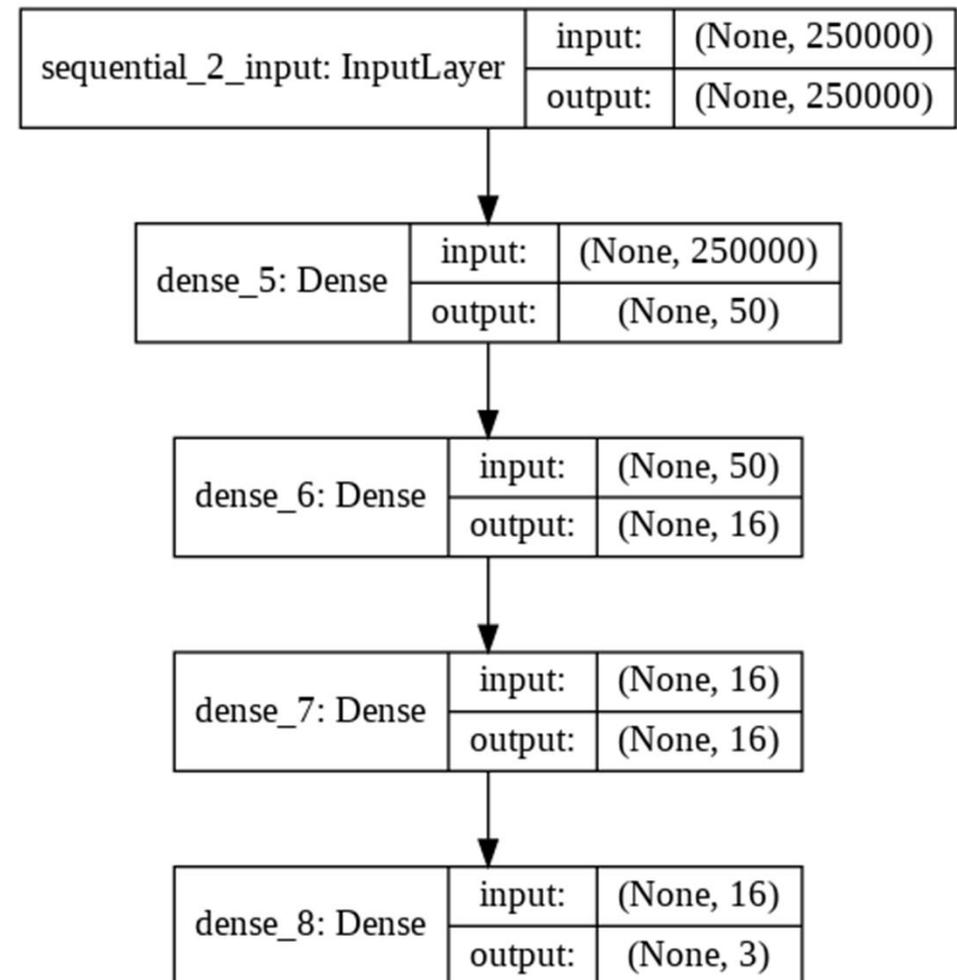


- Entrada da rede neural:
 - 250 mil pontos (250 mil atributos)
- Divisão do banco de dados:
 - Treinamento (60%)
 - Validação (40%)
- Tipo de Perda:
 - Categorical Cross entropy
 - Kullback-Leibler Divergence

Fonte: Aatoria própria

MLP com dados crus

- 3 camadas ocultas:
 - 1ª com 50 neurônios
 - 2ª com 16 neurônios
 - 3ª com 16 neurônios
- Funções de ativação:
 - ReLu nas camadas ocultas
 - Softmax na camada de saída

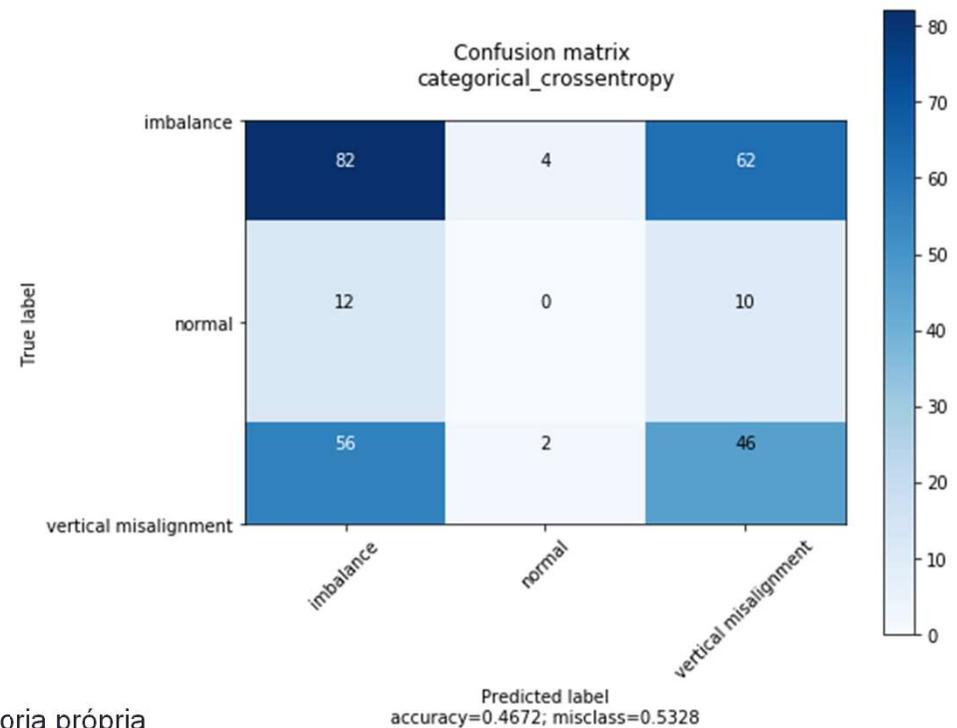
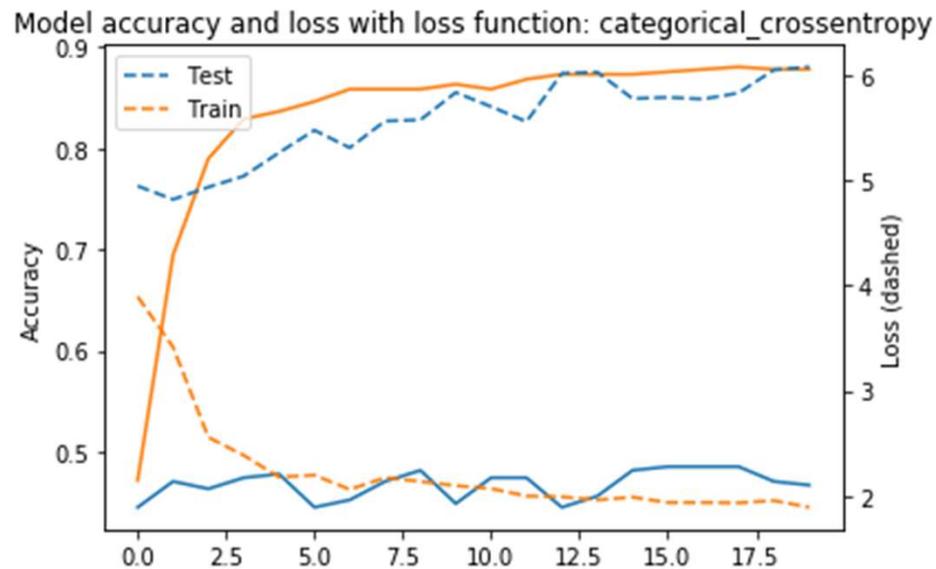


Fonte: Autoria própria

MLP com dados crus

- 20 épocas
- Otimizador Adam
- Perda: Cross entropy
- Pacote: 20
- Pesos iguais

Variação da função de perda



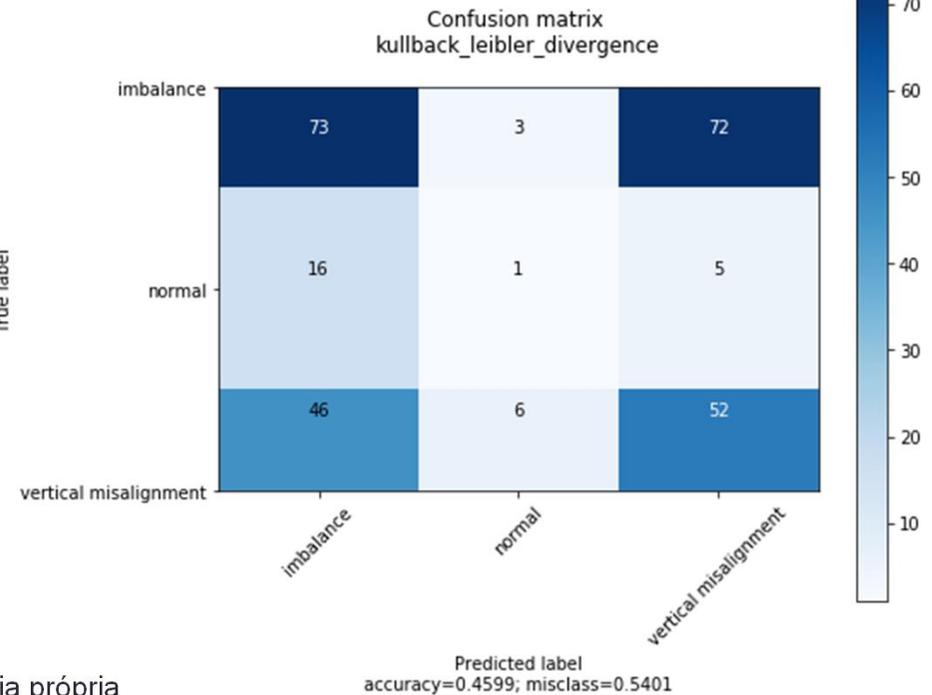
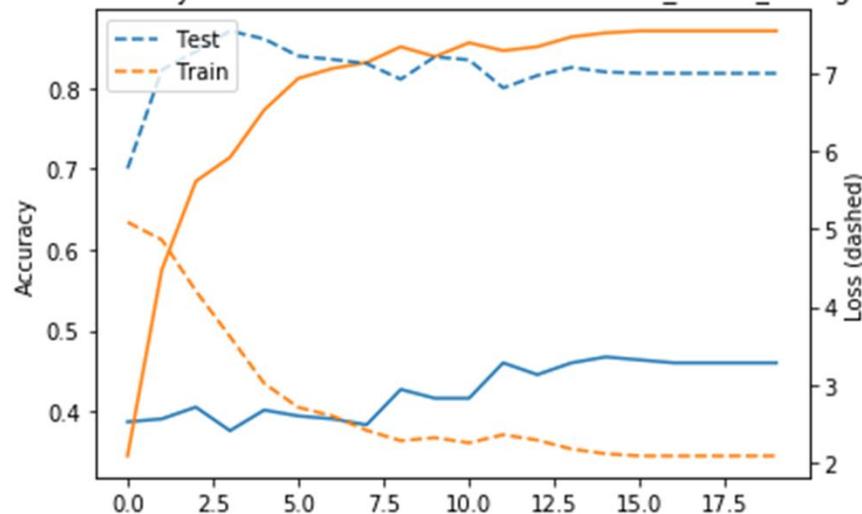
Fonte: Autoria própria

MLP com dados crus

- 20 épocas
- Otimizador Adam
- Perda: Kullback Leibler Divergence
- Pacote: 20
- Pesos iguais

Variação da função de perda

Model accuracy and loss with loss function: kullback_leibler_divergence

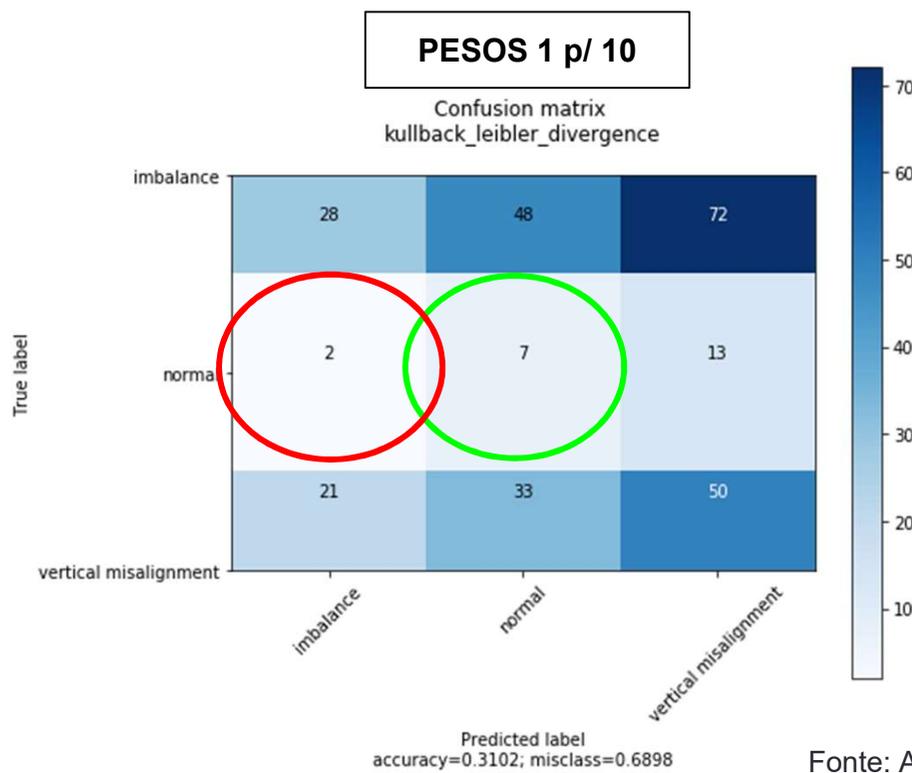


Fonte: Aatoria própria

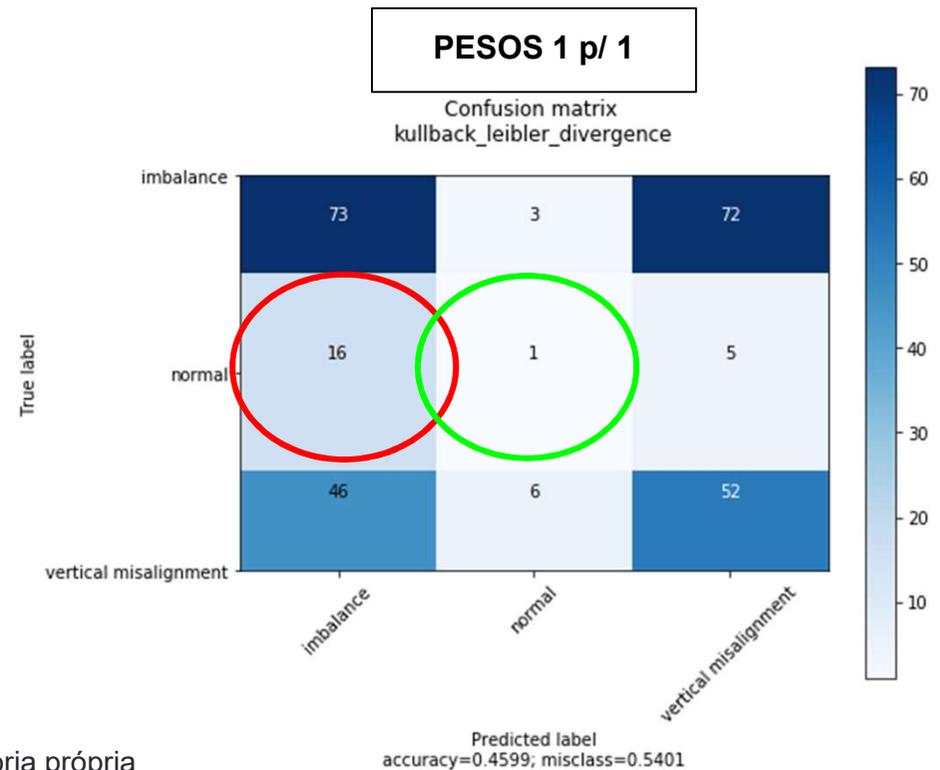
MLP com dados crus

- 20 épocas
- Otimizador Adam
- Perda: Kullback Leibler Divergence
- Pacote: 20

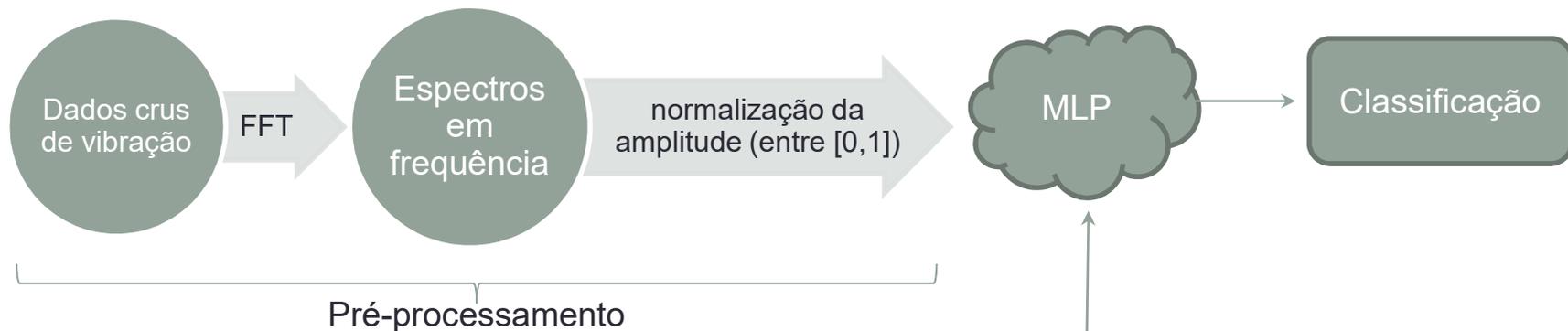
Variação dos pesos das classes



Fonte: Aatoria própria



MLP com espectros de frequência



- **Entrada da rede neural:**
 - Inicialmente: 62.500 atributos
 - Limitamos em 3.000 atributos
- **Divisão do banco de dados:**
 - Treinamento (60%)
 - Validação (40%)
- **Tipo de Perda:**
 - Categorical Cross entropy
 - Kullback Leibler Divergence

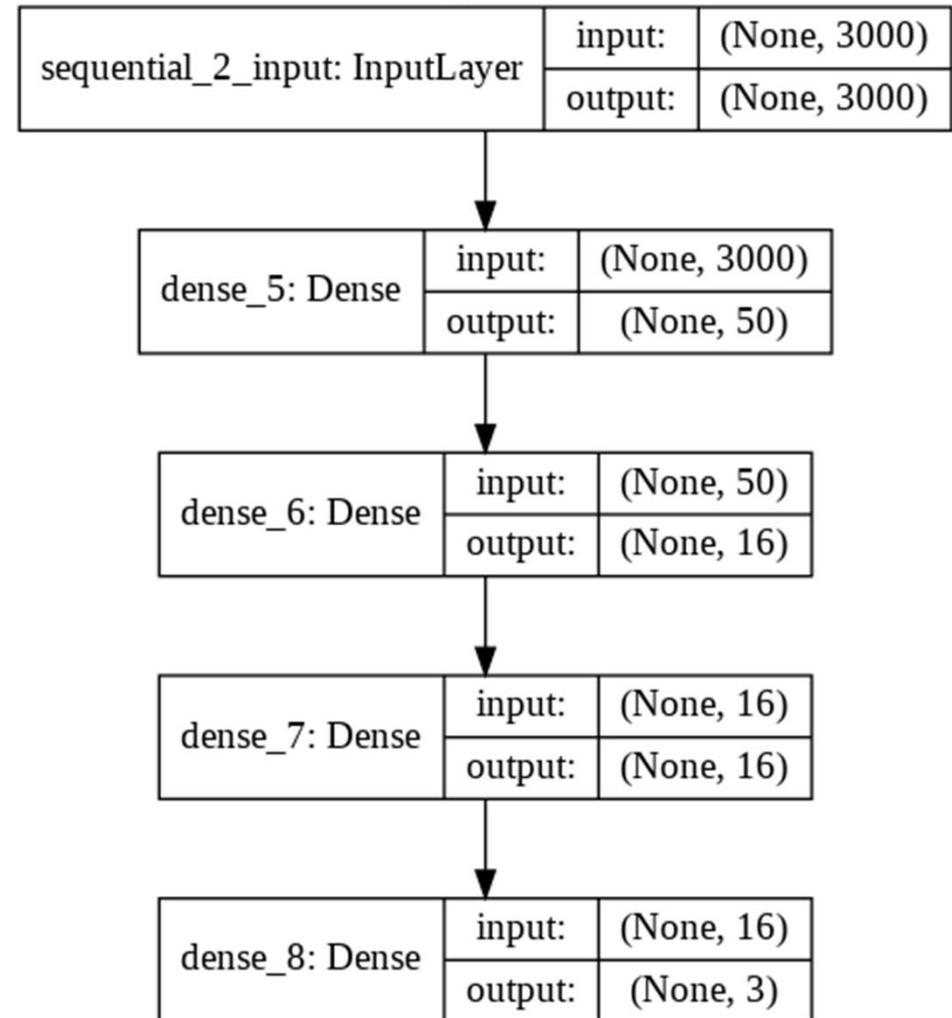
Vetor de classificação:

1. Normal
2. Desbalanceamento
3. Desalinhamento vertical

Fonte: Aatoria própria

MLP com espectros de frequência

- 3 camadas ocultas:
 - 1ª com 50 neurônios
 - 2ª com 16 neurônios
 - 3ª com 16 neurônios
- Funções de ativação:
 - ReLu nas camadas ocultas
 - Softmax na camada de saída

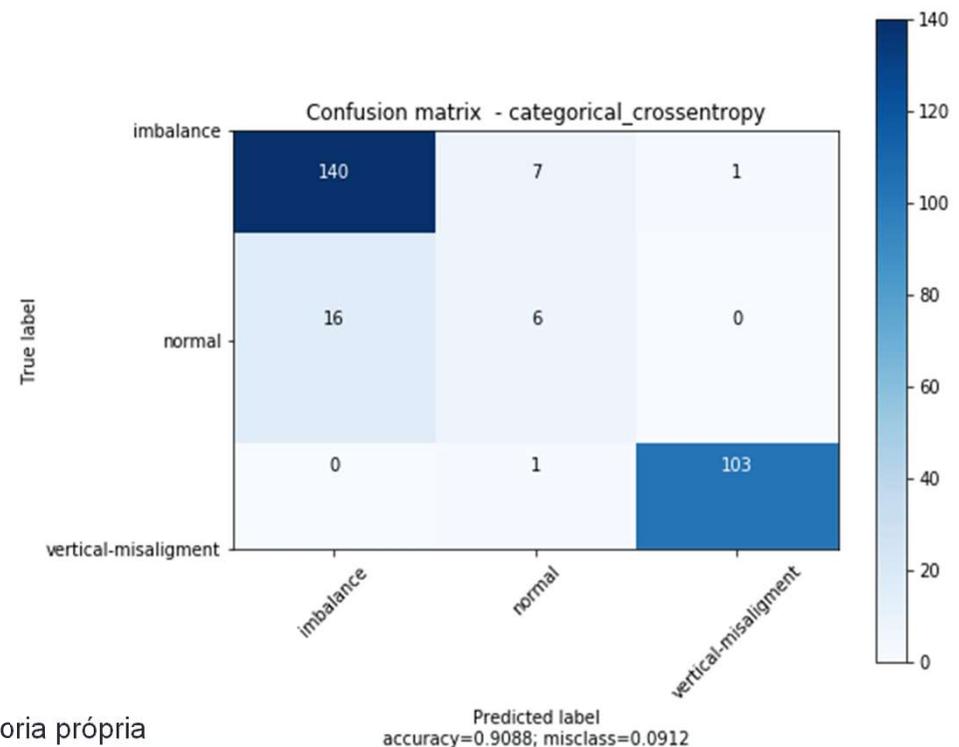
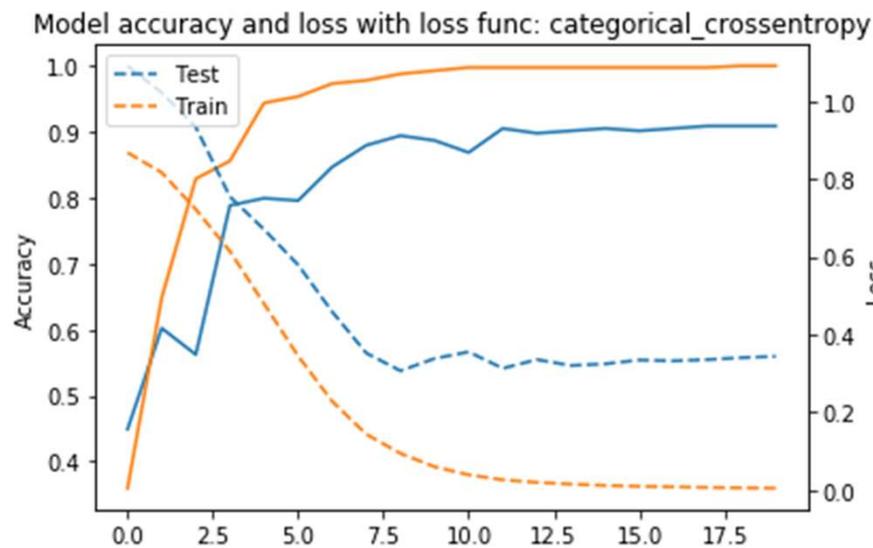


Fonte: Autoria própria

MLP com espectros de frequência

- 20 épocas
- Otimizador Adam
- Perda: Cross entropy
- Pacote: 20
- Pesos iguais

Variação da função de perda



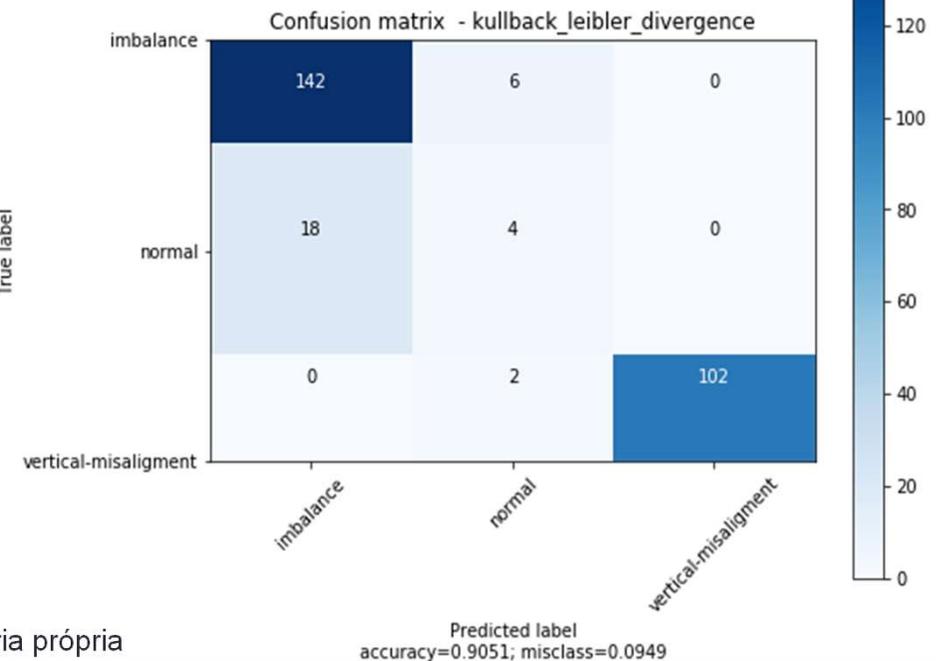
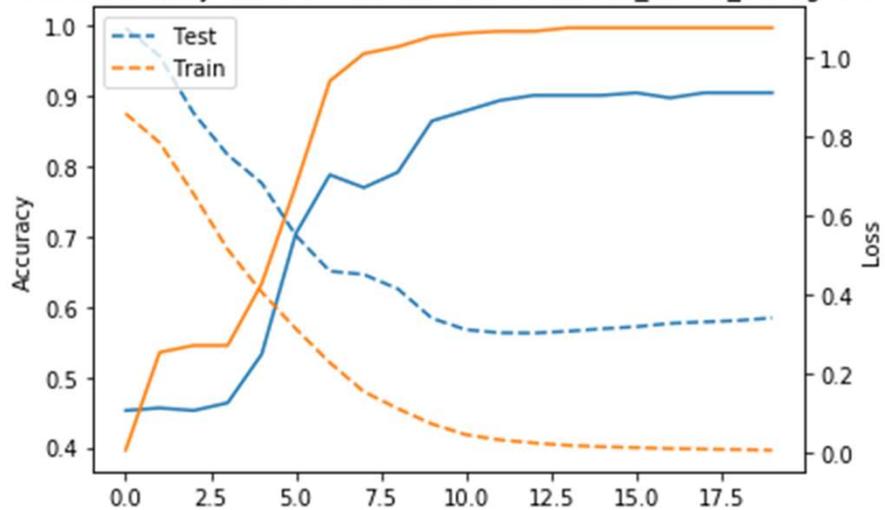
Fonte: Autoria própria

MLP com espectros de frequência

- 20 épocas
- Otimizador Adam
- Perda: Kullback Leibler Divergence
- Pacote: 20
- Pesos iguais

Variação da função de perda

Model accuracy and loss with loss func: kullback_leibler_divergence

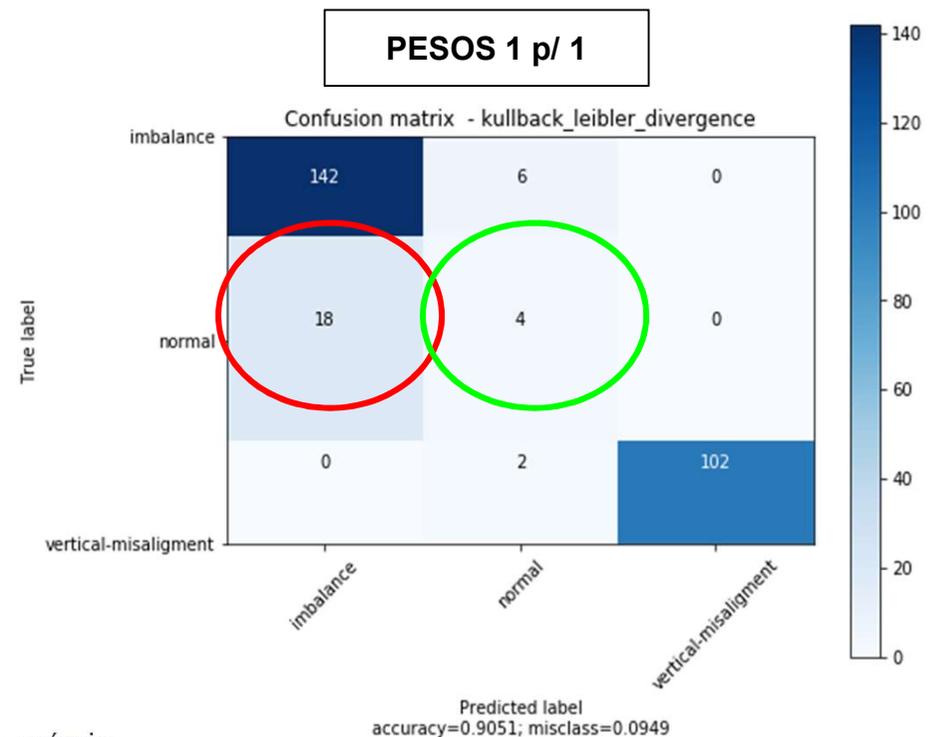
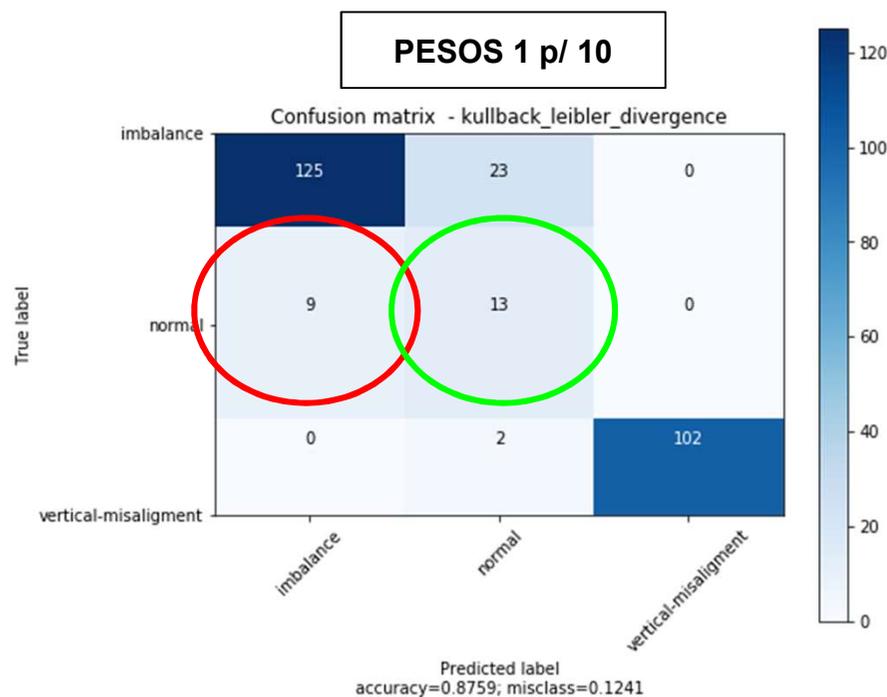


Fonte: Aatoria própria

MLP com espectros de frequência

- 20 épocas
- Otimizador Adam
- Perda: Kullback Leibler Divergence
- Pacote: 20

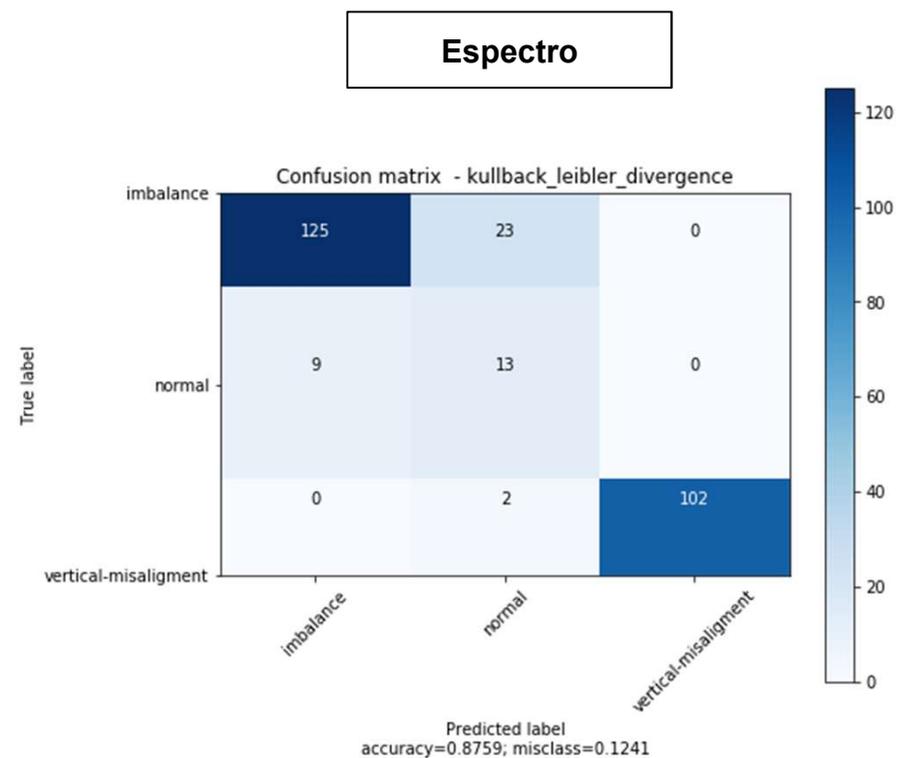
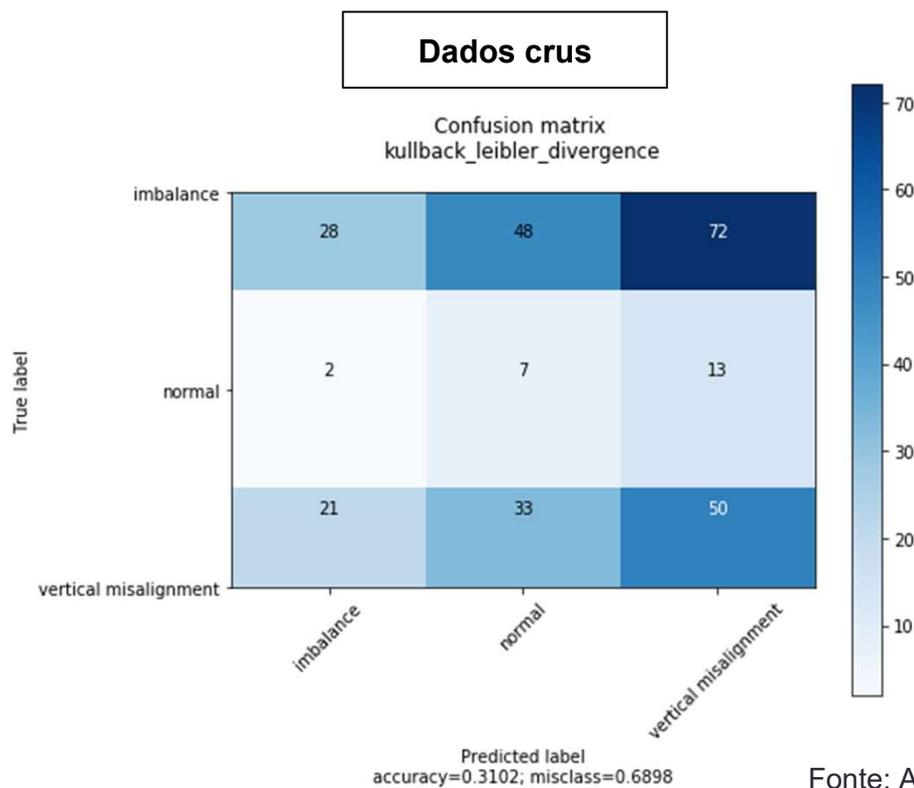
Variação dos pesos das classes



Fonte: Aatoria própria

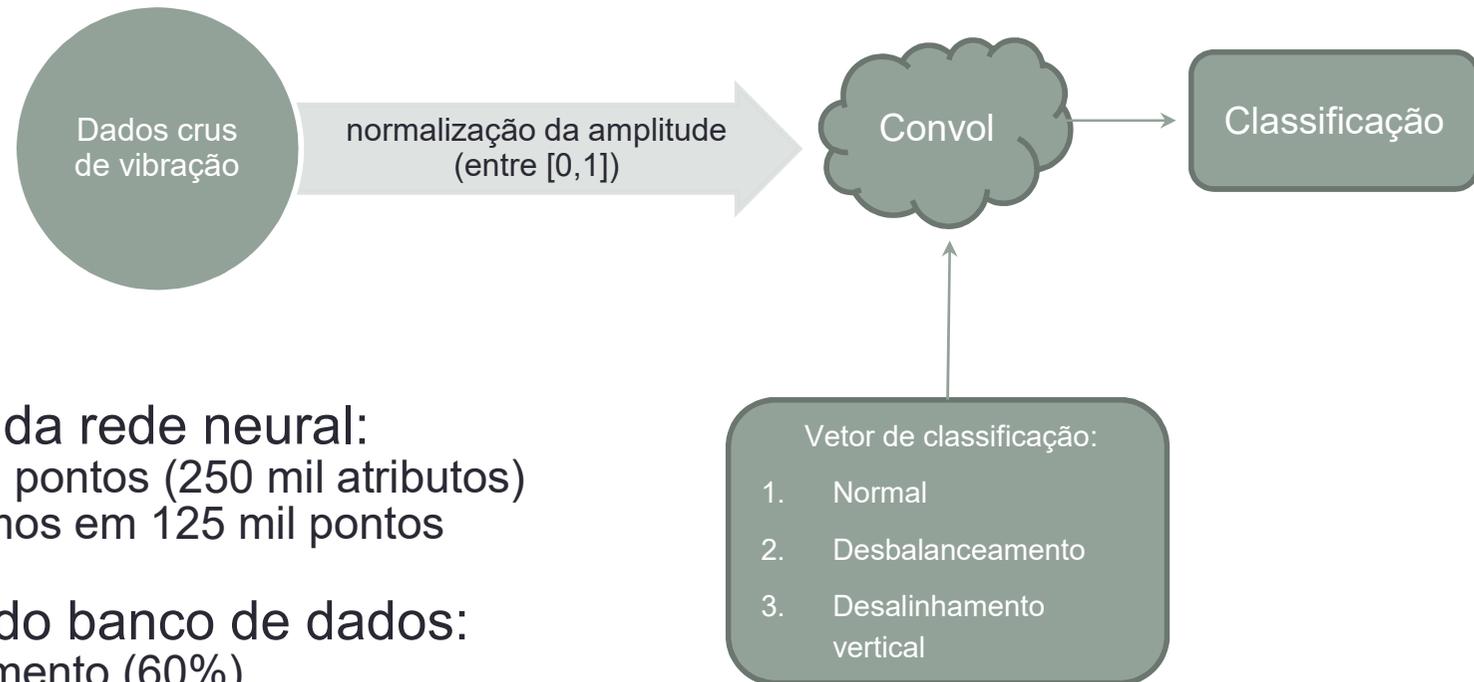
MLP e Processamento de sinais

- Comparação entre resultados usando dados crus e processamento de sinais



Fonte: Autoria própria

Convolutacional com dados crus

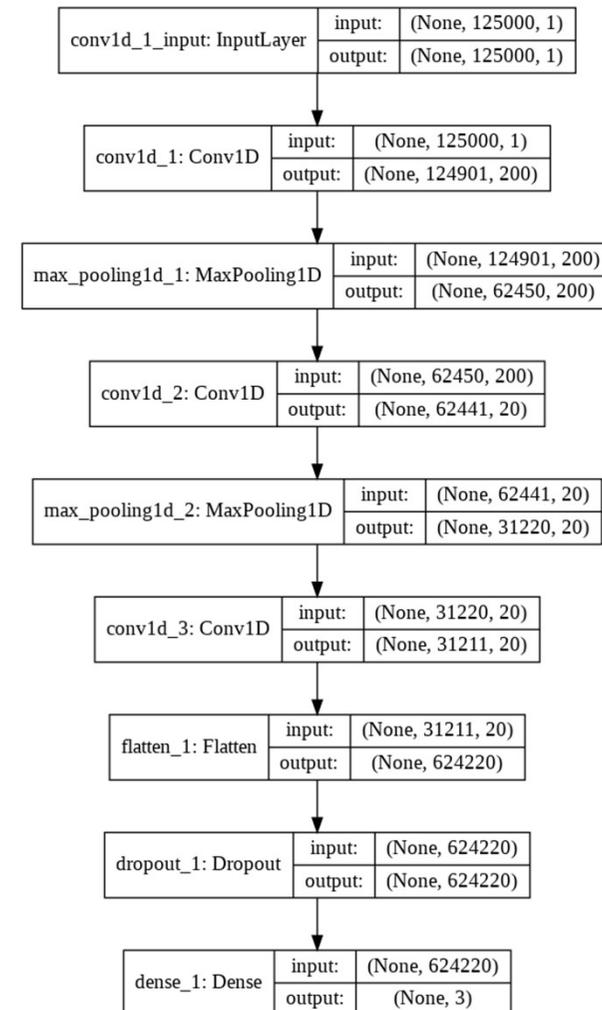


- Entrada da rede neural:
 - 250 mil pontos (250 mil atributos)
 - Limitamos em 125 mil pontos
- Divisão do banco de dados:
 - Treinamento (60%)
 - Validação (40%)
- Tipo de Perda:
 - Categorical Cross entropy
 - Kullback Leibler Divergence

Fonte: Autoria própria

Convolucionais com dados crus

- 1ª camada convolucional:
 - Filtro = 200 e Kernel = 100
- Camada de maxpooling (Pool size = 2)
- 2ª camada convolucional:
 - Filtro = 20 e Kernel = 10
- Camada de maxpooling (Pool size = 2)
- 3ª camada convolucional:
 - Filtro = 20 e Kernel = 10
- Flatten
- Dropout (0,5)
- MLP com 3 saídas
- Funções de ativação:
 - ReLu nas camadas convolucionais
 - Softmax na camada de saída

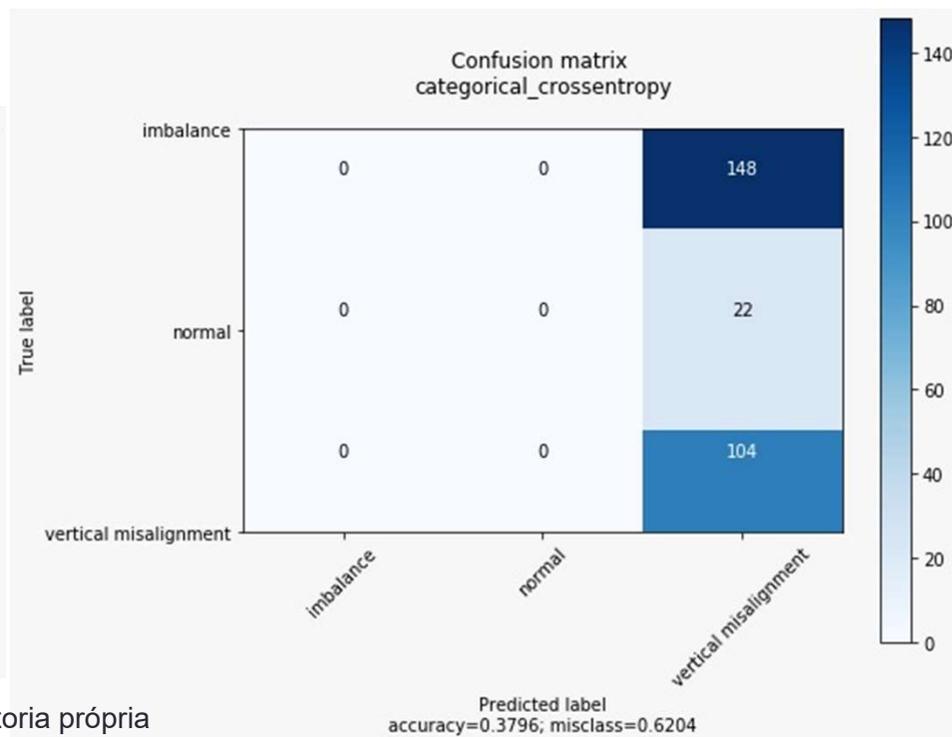
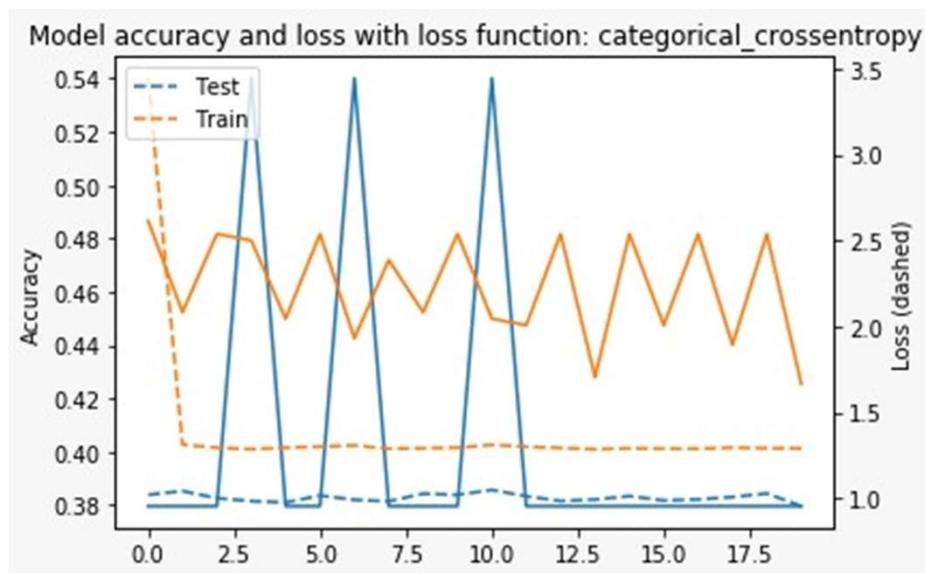


Fonte: Autoria própria

Conv com dados crus

- 20 épocas
- Otimizador Adam
- Perda: Cross entropy
- Pacote: 20
- Pesos 1 para 4

Variação da função de perda

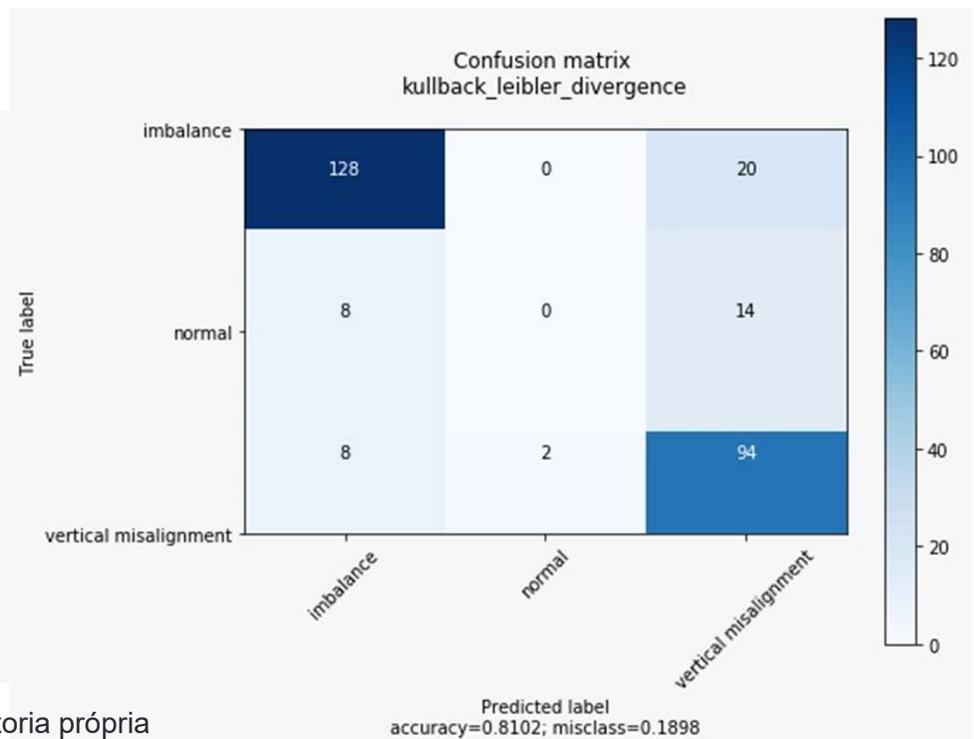
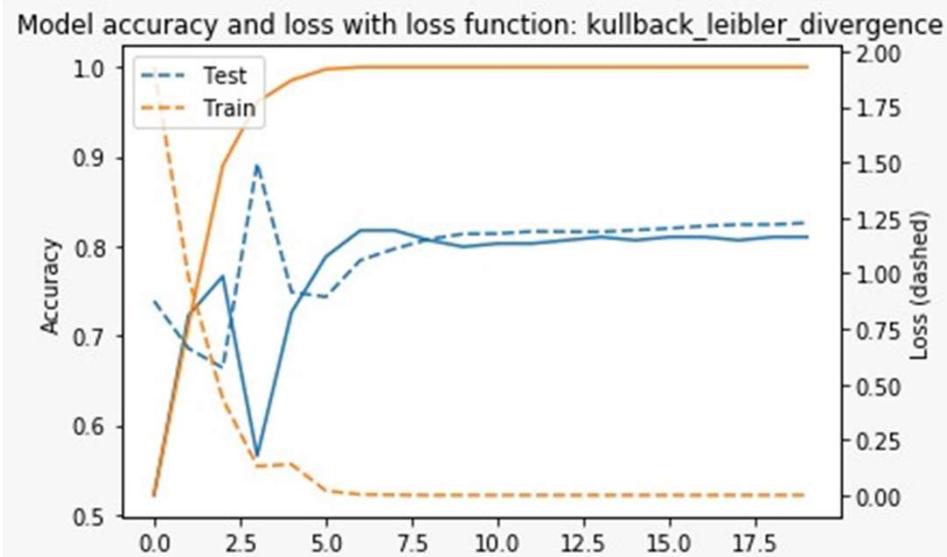


Fonte: Autoria própria

Conv com dados crus

- 20 épocas
- Otimizador Adam
- Perda: Kullback Leibler Divergence
- Pacote: 20
- Pesos 1 para 4

Variação da função de perda

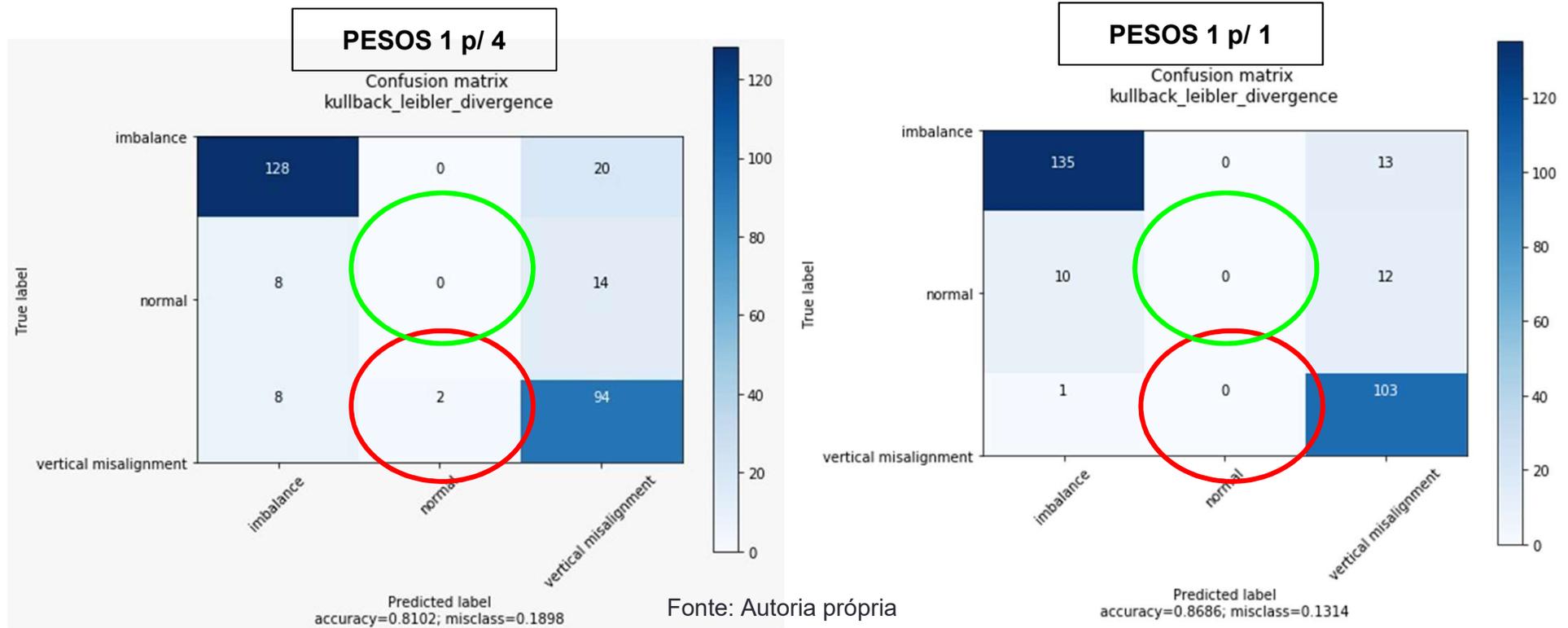


Fonte: Aatoria própria

Conv com dados crus

- 20 épocas
- Otimizador Adam
- Perda: Kullback Leibler Divergence
- Pacote: 20

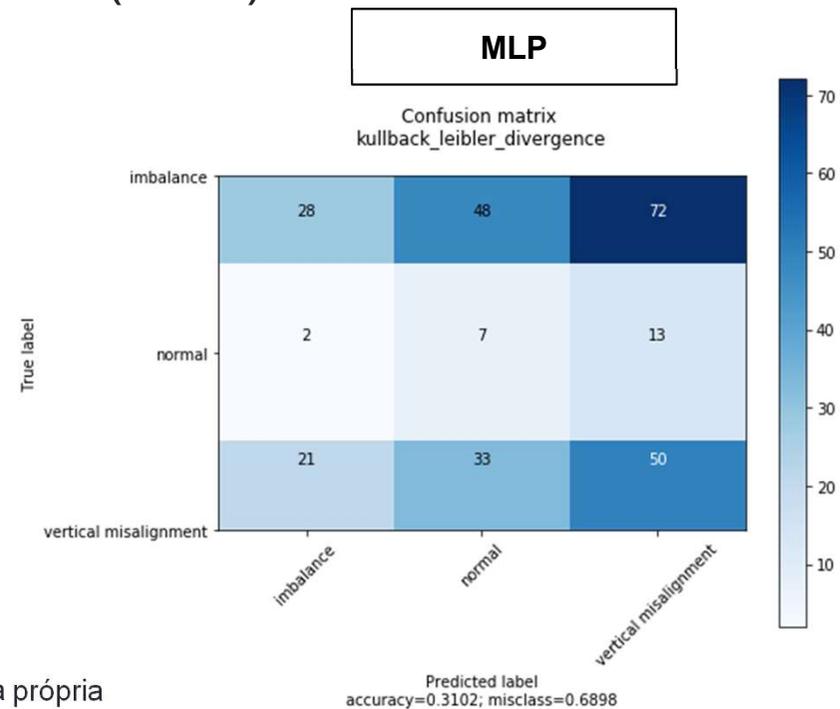
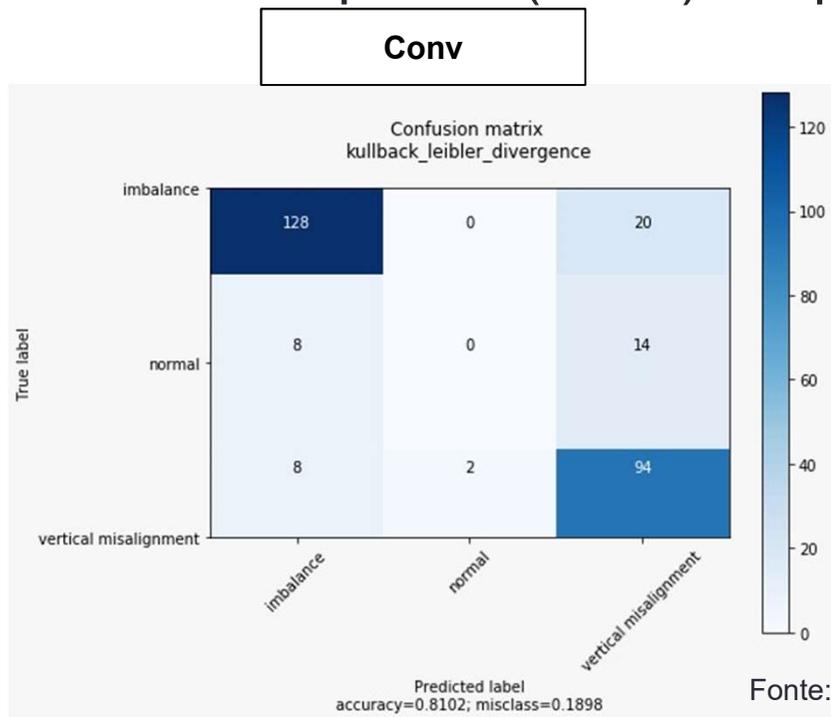
Variação de pesos das classes



Conv E MLP com dados crus

- 20 épocas
- Otimizador Adam
- Perda: Kullback Leibler Divergence
- Pacote: 20
- Pesos: 1 para 4 (Conv) e 1 para 10 (MLP)

CONVOLUCIONAL
TEVE UM
DESEMPENHO
MELHOR NO GERAL

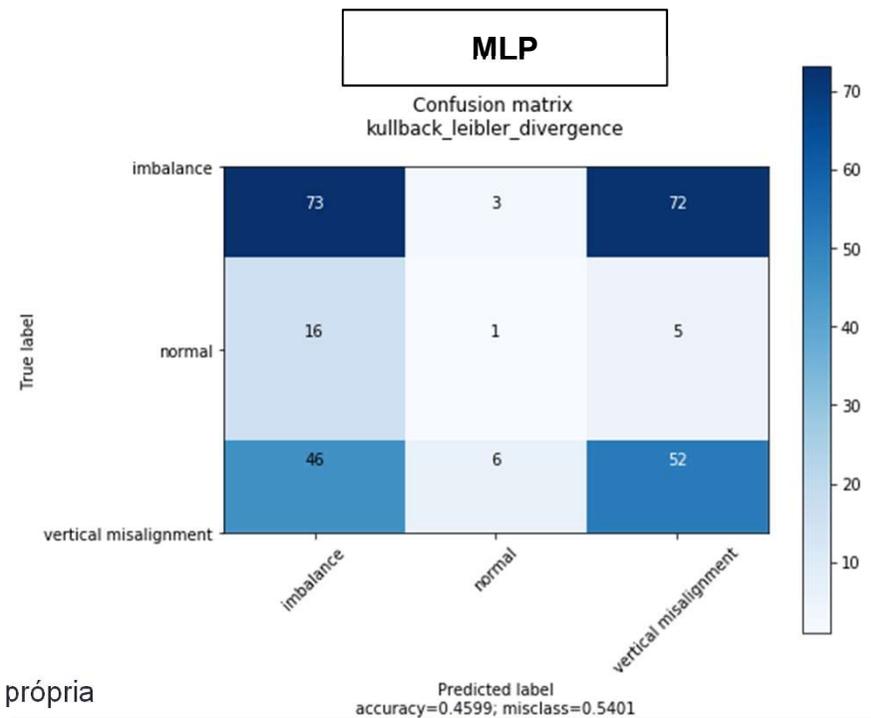
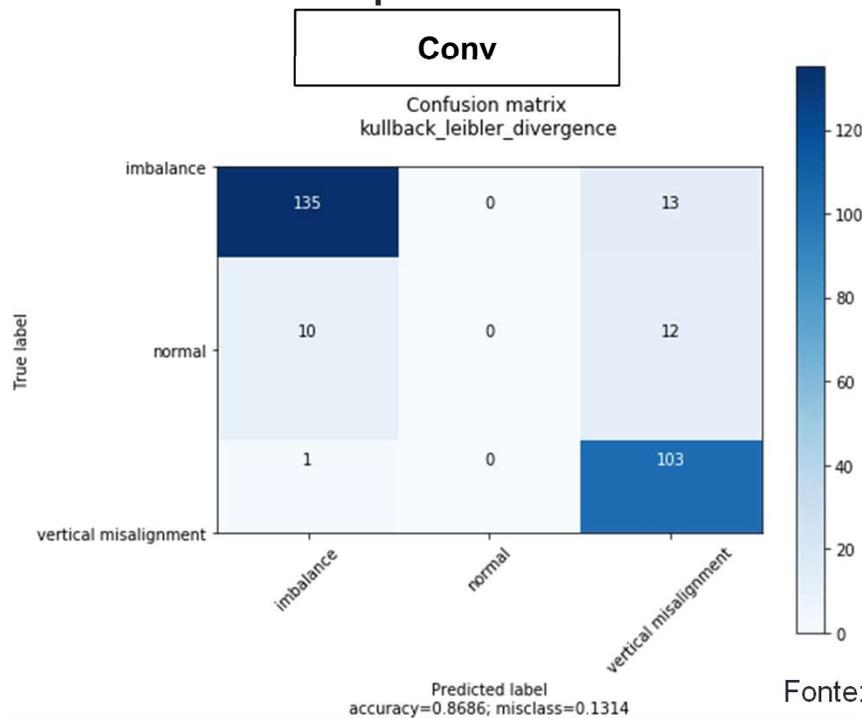


Fonte: Autoria própria

Conv E MLP com dados crus

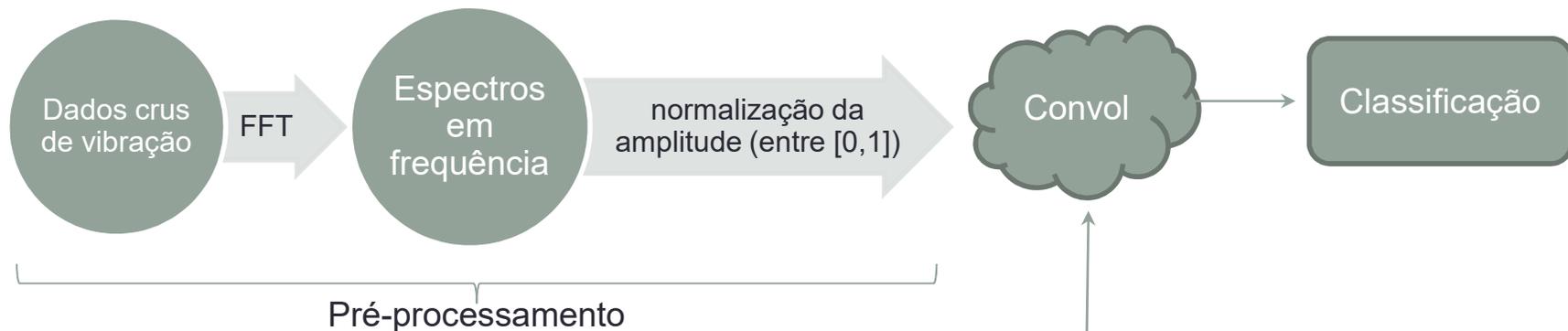
- 20 épocas
- Otimizador Adam
- Perda: Kullback Leibler Divergence
- Pacote: 20
- Pesos: 1 para 1

CONVOLUCIONAL
TEVE UM
DESEMPENHO
MELHOR NO GERAL

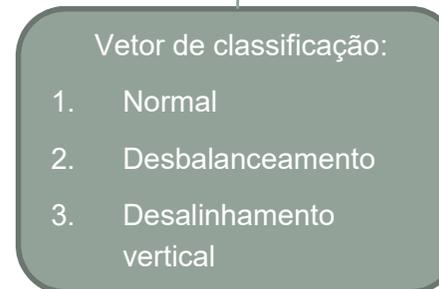


Fonte: Aatoria própria

Convolutacional com espectros



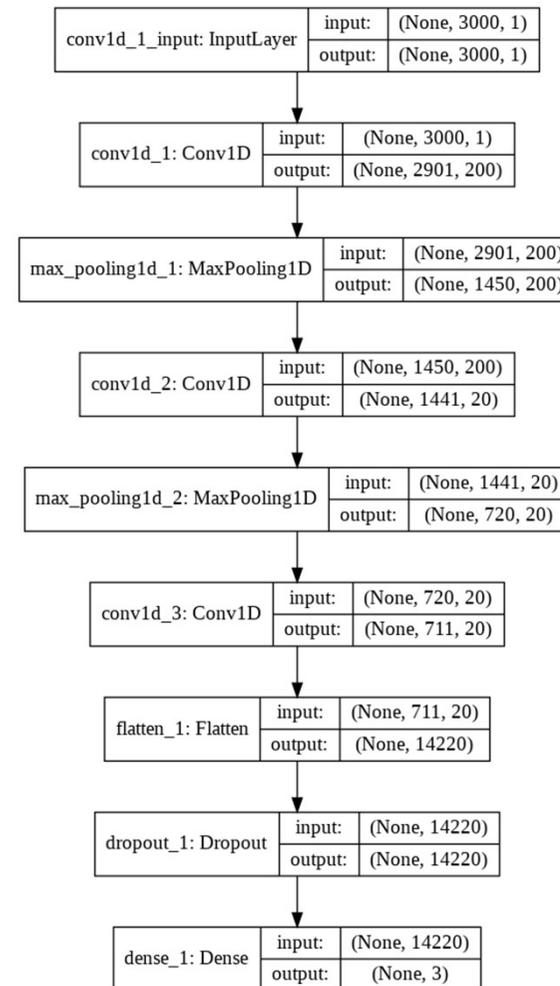
- **Entrada da rede neural:**
 - Inicialmente: 62.500 atributos
 - Limitamos em 3.000 atributos
- **Divisão do banco de dados:**
 - Treinamento (60%)
 - Validação (40%)
- **Tipo de Perda:**
 - Categorical Cross entropy
 - Kullback Leibler Divergence



Fonte: Aatoria própria

Conv com espectros de frequência

- 1ª camada convolucional:
 - Filtro = 200 e Kernel = 100
- Camada de maxpooling (Pool size = 2)
- 2ª camada convolucional:
 - Filtro = 20 e Kernel = 10
- Camada de maxpooling (Pool size = 2)
- 3ª camada convolucional:
 - Filtro = 20 e Kernel = 10
- Flatten
- Dropout (0,5)
- MLP com 3 saídas
- Funções de ativação:
 - ReLu nas camadas convolucionais
 - Softmax na camada de saída

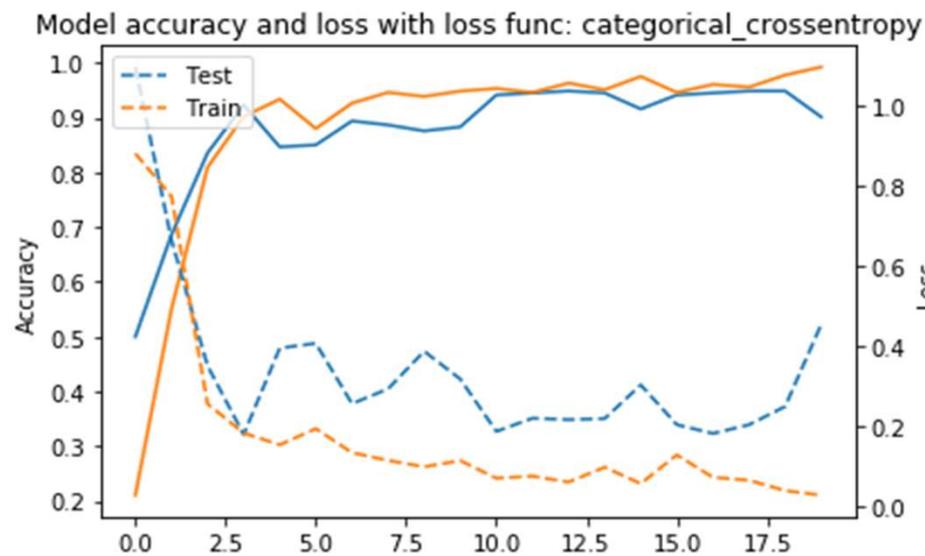


Fonte: Autoria própria

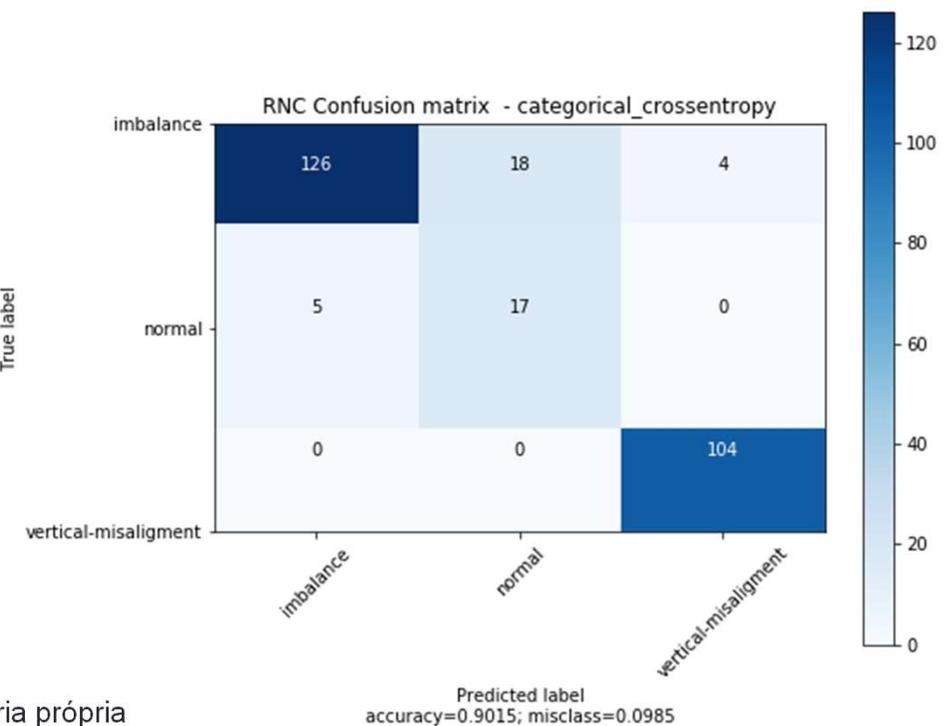
Conv com espectros de frequência

- 20 épocas
- Otimizador Adam
- Perda: Cross entropy
- Pacote: 20
- Pesos 1 para 4

Variação da função de perda



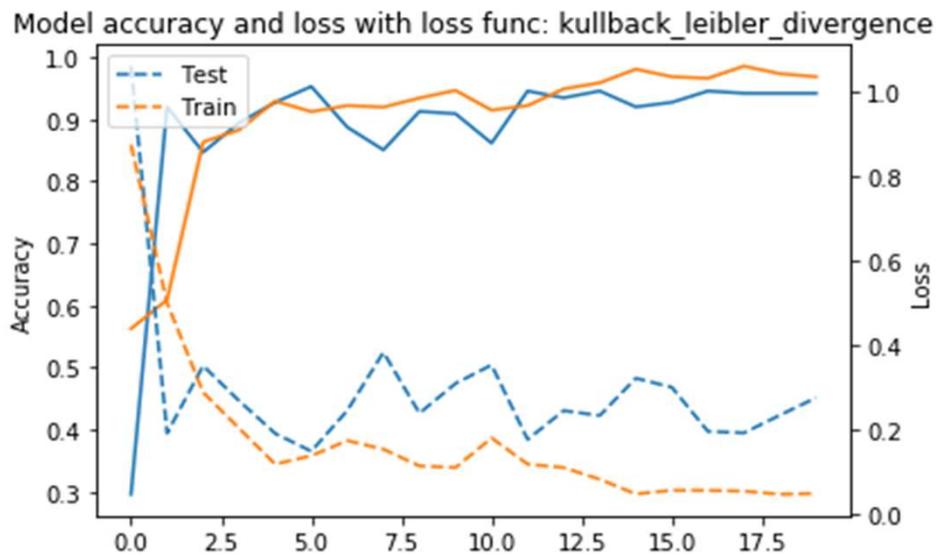
Fonte: Aatoria própria



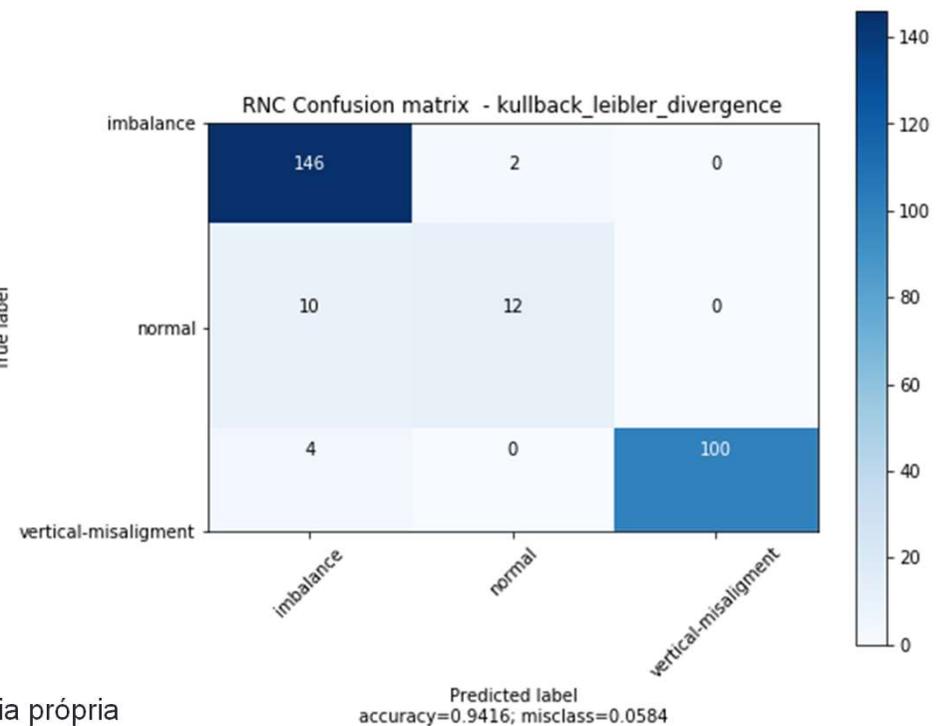
Conv com espectros de frequência

- 20 épocas
- Otimizador Adam
- Perda: Kullback Leibler Divergence
- Pacote: 20
- Pesos 1 para 4

Variação da função de perda



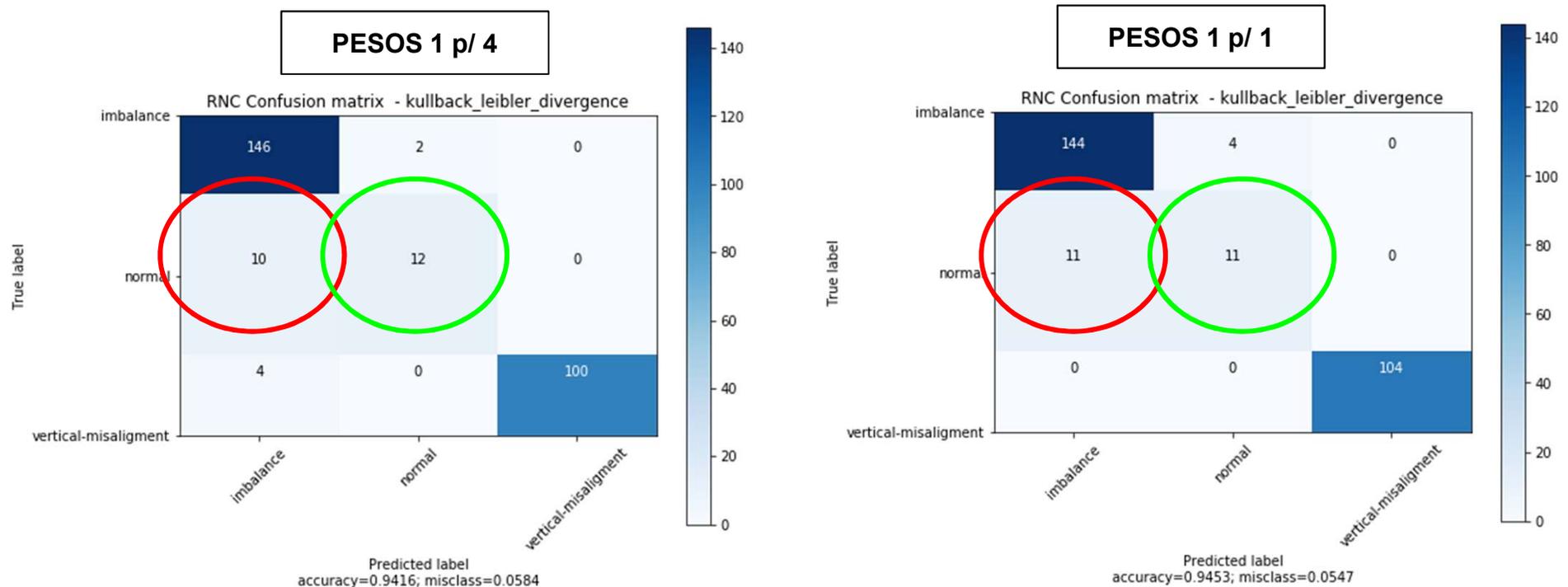
Fonte: Autoria própria



Conv com espectros de frequência

- 20 épocas
- Otimizador Adam
- Perda: Kullback Leibler Divergence
- Pacote: 20

Variação de importância das classes

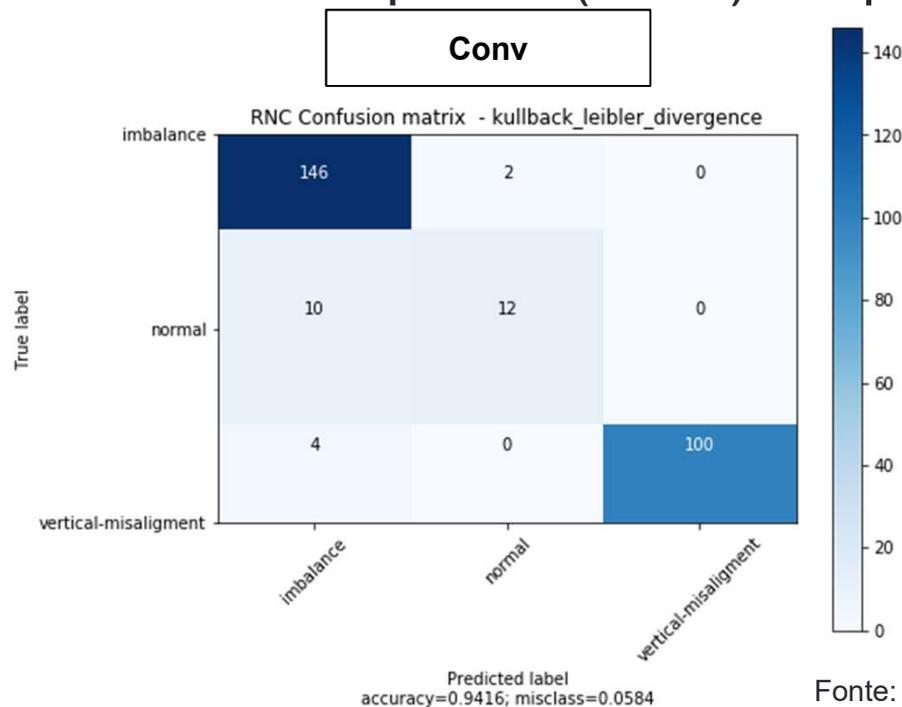


Fonte: Autoria própria

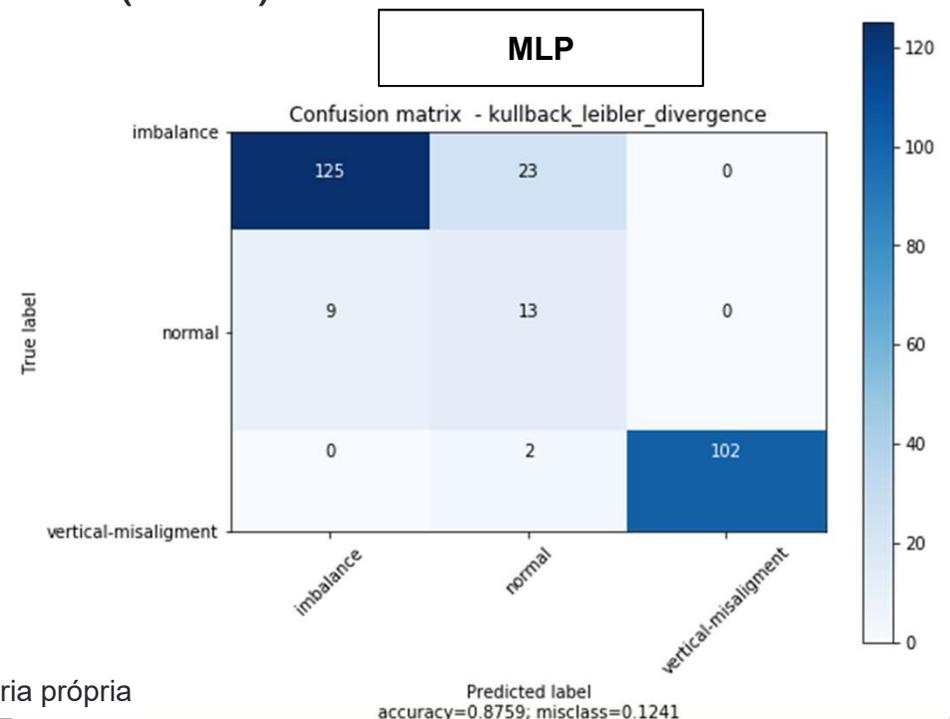
Conv E MLP com espectros

- 20 épocas
- Otimizador Adam
- Perda: Kullback Leibler Divergence
- Pacote: 20
- Pesos 1 para 4 (Conv) e 1 para 10 (MLP)

CONVOLUCIONAL
TEVE UM
DESEMPENHO
MELHOR NO GERAL



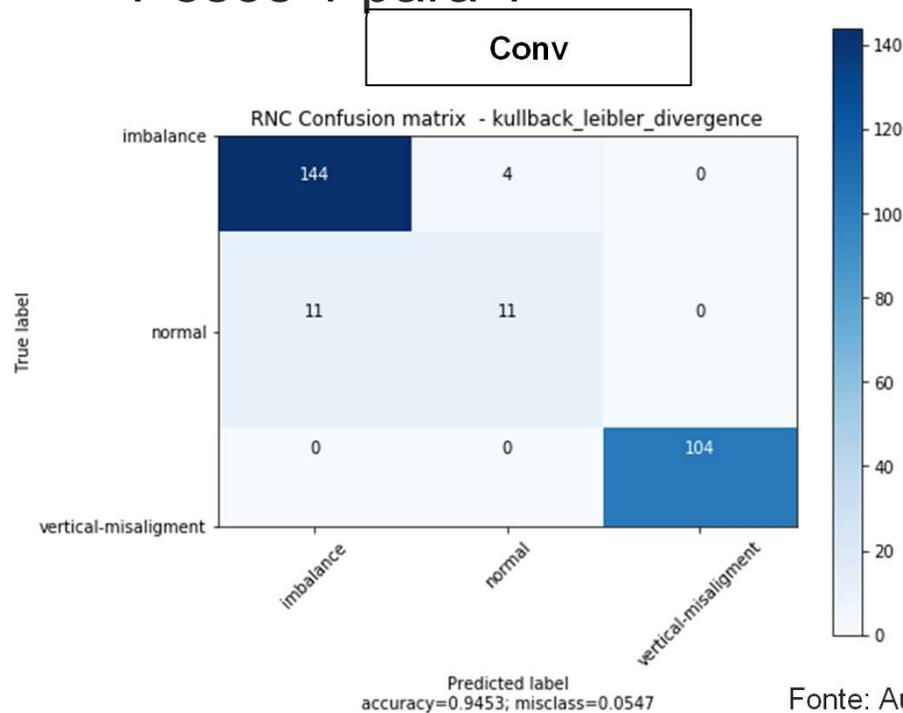
Fonte: Aatoria própria



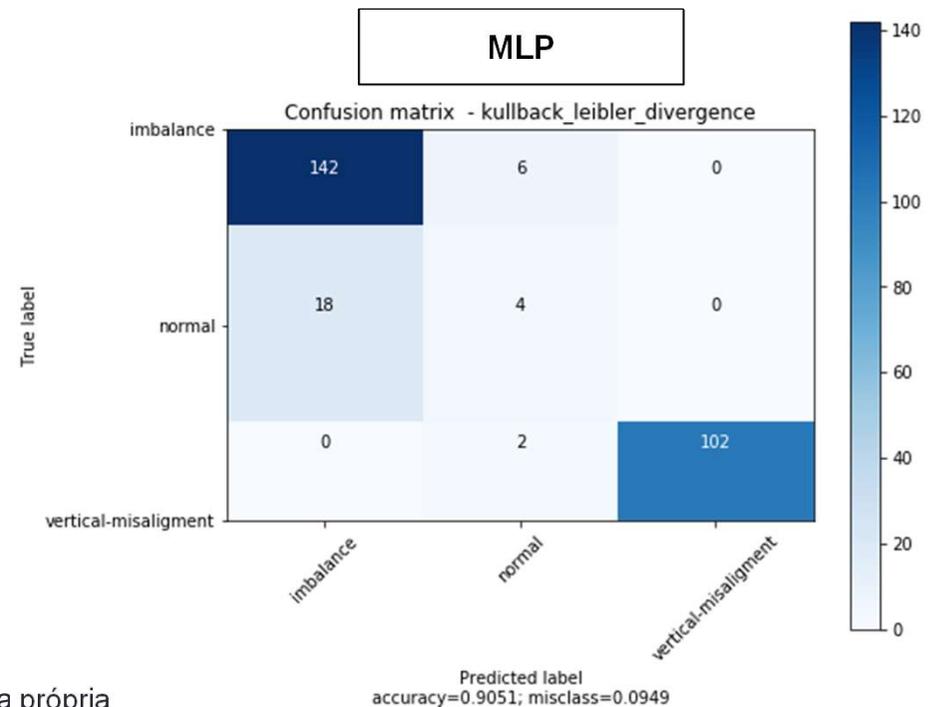
Conv E MLP com espectros

- 20 épocas
- Otimizador Adam
- Perda: Kullback Leibler Divergence
- Pacote: 20
- Pesos 1 para 1

A diferença de acurácia aumenta quando comparamos as arquiteturas com PESO 1 para 1

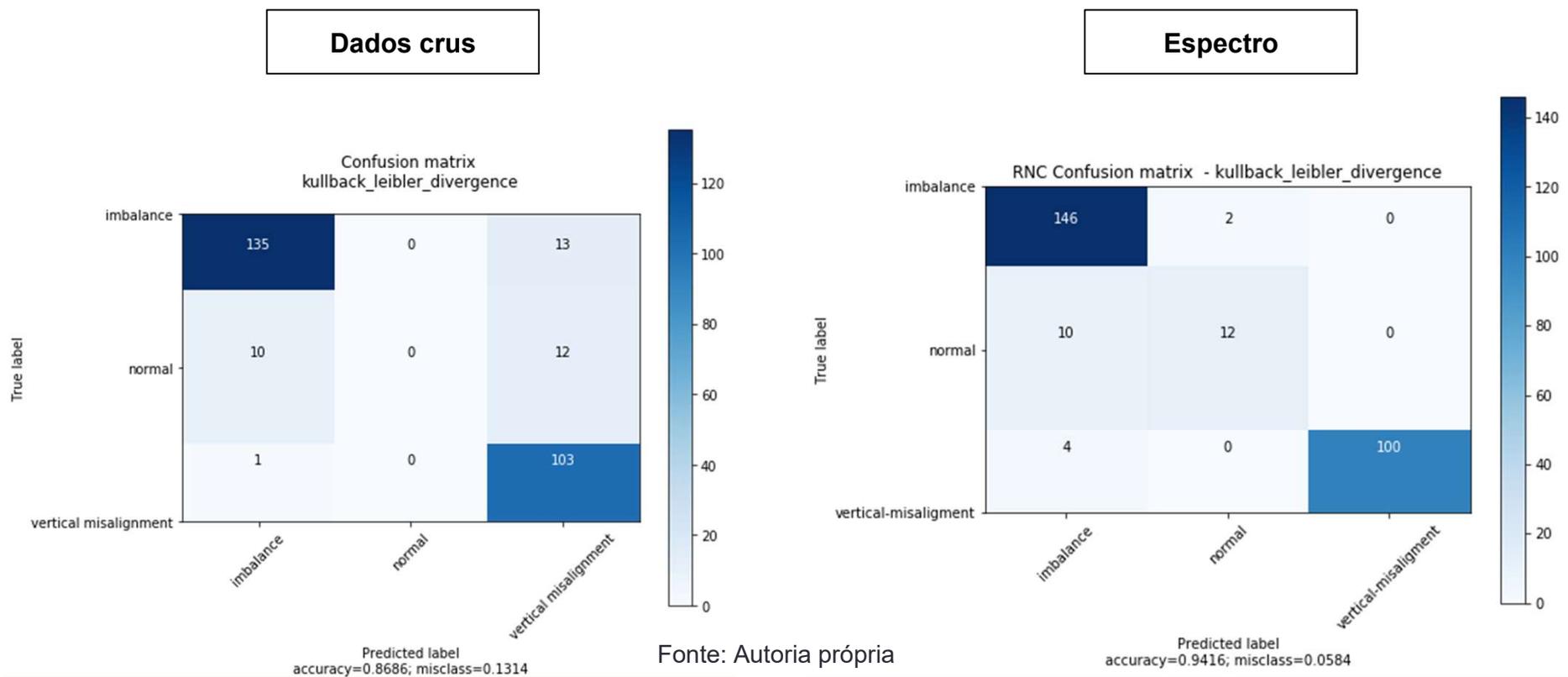


Fonte: Aatoria própria



Conv e Processamento de sinais

- Comparação entre resultados usando dados crus e processamento de sinais



CONCLUSÕES

Comparação entre modelos

	Tipo de dados	Função de perda	Acurácia
MLP	Dados crus	Cross Entropy	46,72%
		KL divergence	45,99%
		KL div + pesos	31,02%
	Dados em espectro de frequência	Cross Entropy	90,88%
		KL divergence	90,51%
		KL div + pesos	87,59%
Convencionais	Dados crus	Cross Entropy	37,96%
		KL divergence	86,86%
		KL div + pesos	81,02%
	Dados em espectro de frequência	Cross Entropy	90,15%
		KL divergence	94,53%
		KL div + pesos	94,16%

Fonte: Autoria própria

Conclusões

- MLP não classifica dados crus
 - Classificação próxima à aleatória
 - Muitos nós para treinar e poucos dados disponíveis
- Redes convolucionais apresentaram potencialidade de classificar dados crus
- Transformada de Fourier funciona como um “feature extraction”
 - Redução de dimensão
 - Concentração de informação
- Seleção da função de perda causa grande impacto no resultado

Conclusões

- Redes treinadas com dados desbalanceados classificam menos as classes com poucos dados
 - Resultado pior para MLP
- Redes convolucionais apresentaram desempenho melhor no geral
 - Realizou análise gráfica dos espectros de frequência
 - Mesmo comportamento que análises clássicas de processamento de sinais buscam: aumento dos harmônicos, das chamadas frequências de banda lateral e etc;
 - Devido à sua arquitetura, Redes Convolucionais tiveram menor número de nós que as MLPs.

Trabalhos futuros

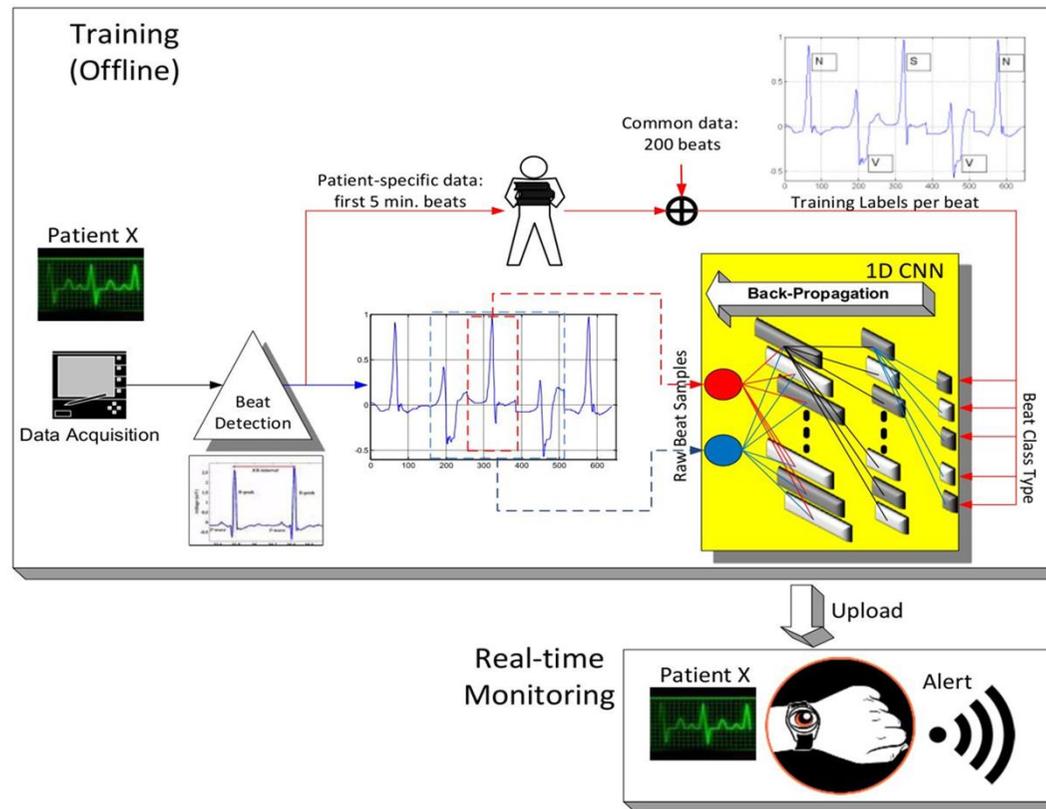
- Há espaço para melhoria com varreduras de tamanhos de pacotes, quantidade de épocas, otimizadores, balanceamento dos dados.
- Investigar a potencialidade das Redes Convolucionais na classificação de dados brutos
- Repetir os ensaios utilizando o espectro em frequência na escala de ordem de harmônico, ao invés de na escala de Hz

- $ordem = \frac{frequência(x)}{rotação}$

OUTRAS APLICAÇÕES

Monitor de eletrocardiograma

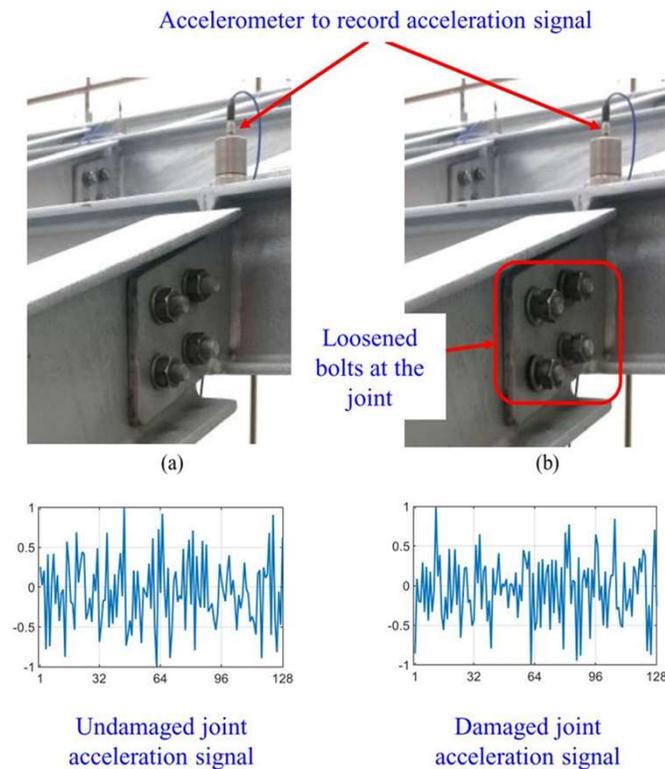
- Monitor de eletrocardiograma em tempo real



Fonte: (KIRANYAZ et al., 2019)

Detecção de desgastes em estruturas

- Detecção de desgastes em estruturas de construção civil por meio de medição de vibração



Fonte: (KIRANYAZ et al., 2019)

Referências

- BARÃO, Renato Casali. **Algoritmos de controle ótimo quadrático com restrições**, 1997.
- CHEN, ZhiQiang; LI, Chuan; SANCHEZ, René-Vinicio. **Gearbox Fault Identification and Classification with Convolutional Neural Networks**. *Shock and Vibration*, v. 2015, n. 2, p. 1–10, 2015. doi:10.1155/2015/390134.
- DANIEL GODOY. **Uma explicação visual para função de custo “binary cross-entropy” ou “log loss”**. Disponível em: <<https://medium.com/ensina-ai/uma-explica%C3%A7%C3%A3o-visual-para-fun%C3%A7%C3%A3o-de-custo-binary-cross-entropy-ou-log-loss-eaee662c396c>>. Acesso em: 25 nov. 2019.
- DATA SCIENCE ACADEMY. **Deep Learning Book**. Disponível em: <<http://deeplearningbook.com.br/>>. Acesso em: 25 nov. 2019.
- **Documentação online da biblioteca Keras**. Disponível em: <<https://keras.io/>>. Acesso em: 25 nov. 2019.
- FERREIRA, Martha Dais. **Designing convolutional neural network architectures based on dynamical system concepts**, 2019. Tese (Doutorado em Ciências de Computação e Matemática Computacional) - Universidade de São Paulo. Instituto de Ciências Matemáticas e de Computação. São Carlos, 2019. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-26042019-082539/>>. Acesso em: 19 out. 2019.
- GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. Cambridge: MIT Press, 2017. xxii, 775 p. ISBN 978-0262035613. Disponível em: <<http://www.deeplearningbook.org/>>. Acesso em: 25 nov. 2019.
- HAYKIN, Simon; ENGEL, Paulo Martins. **Redes neurais: Princípios e prática**. 2. ed. Porto Alegre: Bookman, 2001. 900 p. ISBN 9788573077186.
- JANSSENS, Olivier et al. **Convolutional Neural Network Based Fault Detection for Rotating Machinery**. *Journal of Sound and Vibration*, v. 377, p. 331–345, 2016. doi:10.1016/j.jsv.2016.05.027.
- JARRETT, Kevin et al. **What is the best multi-stage architecture for object recognition? 2009 IEEE 12th International Conference on Computer Vision**. 2009 IEEE 12th International Conference on Computer Vision (ICCV). Kyoto: IEEE, 2009. ISBN 978-1-4244-4420-5, p. 2146–2153.

Referências

- JASON BROWNLEE. **Machine Learning Mastery**. Disponível em: <<https://machinelearningmastery.com/>>. Acesso em: 25 nov. 2019.
- KINGMA, Diederik P.; BA, Jimmy. **Adam: A Method for Stochastic Optimization**, 2014. Disponível em: <<http://arxiv.org/pdf/1412.6980v9>>.
- KIRANYAZ, Serkan et al. **1D Convolutional Neural Networks and Applications: A Survey**, 2019. 20 p. Disponível em: <<http://arxiv.org/pdf/1905.03554v1>>.
- MAFAULDA. **MaFaulDa - Machinery Fault Database [Online]**. Disponível em: <http://www02.smt.ufrj.br/~offshore/mfs/page_01.html>. Acesso em: 25 nov. 2019.
- MARINS, Matheus A. et al. **Improved similarity-based modeling for the classification of rotating-machine failures**. *Journal of the Franklin Institute*, v. 355, n. 4, p. 1913–1930, 2018. doi:10.1016/j.jfranklin.2017.07.038.
- PESTANA-VIANA, Denys et al. **The influence of feature vector on the classification of mechanical faults using neural networks**. **2016 IEEE 7th Latin American Symposium on Circuits & Systems (LASCAS)**. 2016 IEEE 7th Latin American Symposium on Circuits & Systems (LASCAS). Florianopolis, Brazil. 27/02/2016 - 01/03/2016: IEEE, 27 fev. 2016 - 1 mar. 2016. ISBN 978-1-4673-7835-2. Disponível em: <<http://www02.smt.ufrj.br/~eduardo/papers/ci116.pdf>>, p. 115–118.
- RAVINDRA PARMAR. **Common Loss functions in machine learning**. Disponível em: <<https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23>>. Acesso em: 25 nov. 2019.
- SHIVA VERMA. **Understanding Input Output shapes in Convolution Neural Network | Keras**. Disponível em: <<https://towardsdatascience.com/understanding-input-and-output-shapes-in-convolution-network-keras-f143923d56ca>>. Acesso em: 25 nov. 2019.

Referências

- WANG, Huaqing et al. **A novel convolutional neural network based fault recognition method via image fusion of multi-vibration-signals.** *Computers in Industry*, v. 105, p. 182–190, 2019. doi:10.1016/j.compind.2018.12.013.
- YANG, Kai et al. **A Blind Spectrum Sensing Method Based on Deep Learning.** *Sensors (Basel, Switzerland)*, v. 19, n. 10, 2019. doi:10.3390/s19102270.