



Escola Politécnica da USP
PSI - Departamento de Engenharia de
Sistemas Eletrônicos

PSI2672 -Apresentação final e demonstração de resultados

Carlos Augusto Bergfeld Mori
Henrique Seiti Ogawa
Pedro Henrique Itio Hayashi

PSI2672 - EPUSP 2015 - Práticas em Reconhecimento de Padrões, Modelagem e
Neurocomputação
Prof. Dr. Emilio Del Moral Hernandez

São Paulo, 24 de Junho de 2015



Carlos Augusto Bergfeld Mori
carlos.mori@usp.br

Reconhecedor de Acordes



Henrique Seiti Ogawa
henrique.ogawa@usp.br

Reconhecedor de Caracteres



Pedro Henrique Itio Hayashi
pedro.hayashi@usp.br

Benchmark de CPUs

1. Benchmark de CPUs - Regressor de Desempenho
2. Reconhecimento de caracteres
3. Reconhecimento de acordes

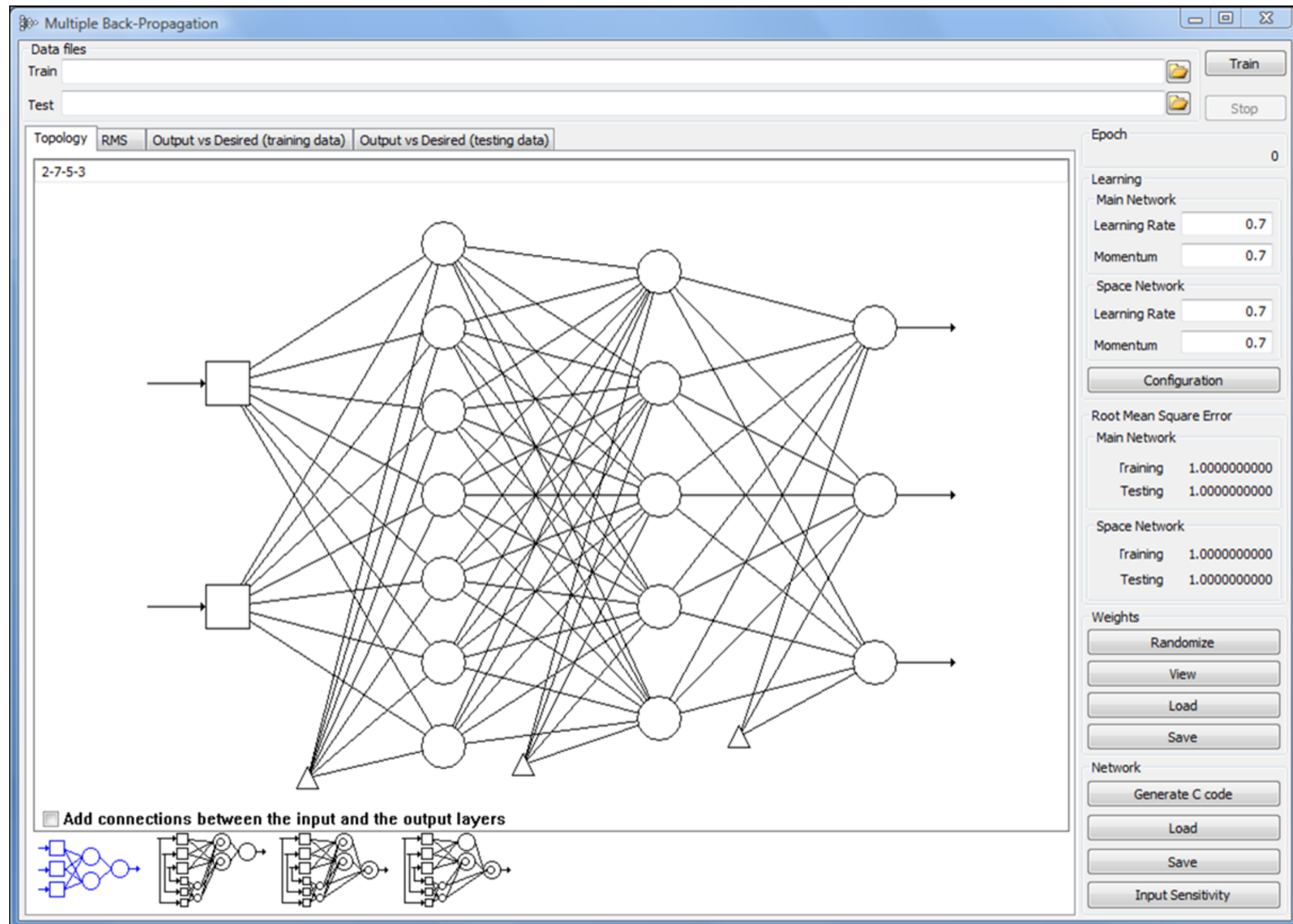
Softwares utilizados

Multiple Back-Propagation

Toolbox de redes neurais do MATLAB

Softwares utilizados

Multiple Back-Propagation

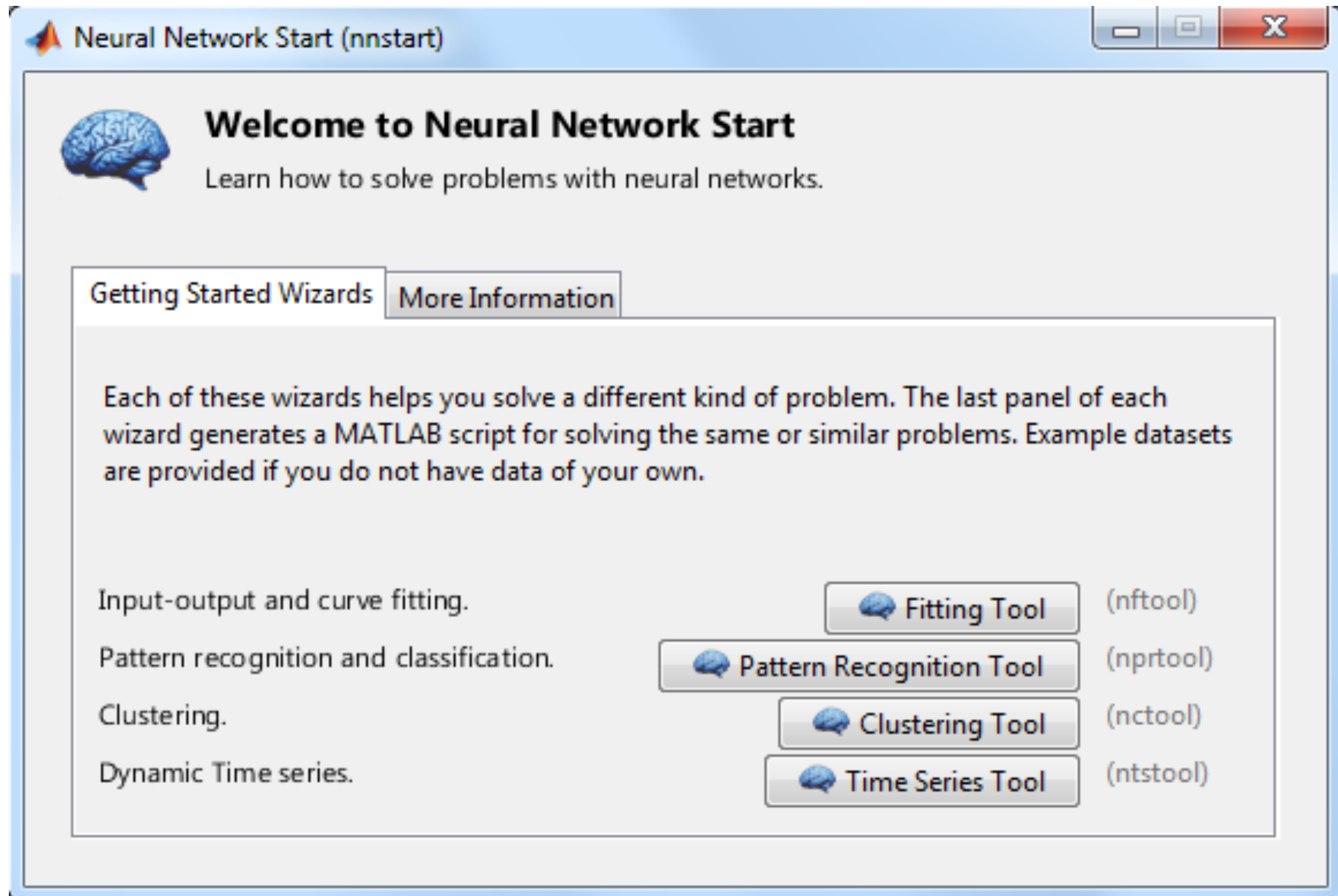


- Implementação mais simples de redes neurais
 - Definição da topologia
 - Entrada dos arquivos de treino e teste
- Treino de redes neurais através do algoritmo Back-Propagation
- Treino de redes mais complexas com o algoritmo Multiple Back-Propagation
(Não estudadas neste curso)

- Salvar a rede: exportar um código C ou salvar pesos
- Treino de uma rede por vez com mudança de parâmetros manual
- Sem medidas de qualidade específicas para classificadores
(Matriz de Confusão, Curva ROC, ...)
- Demanda de processamento alta para redes maiores
 - Número elevado de entradas
 - Número elevado de nós na camada escondida

Softwares utilizados

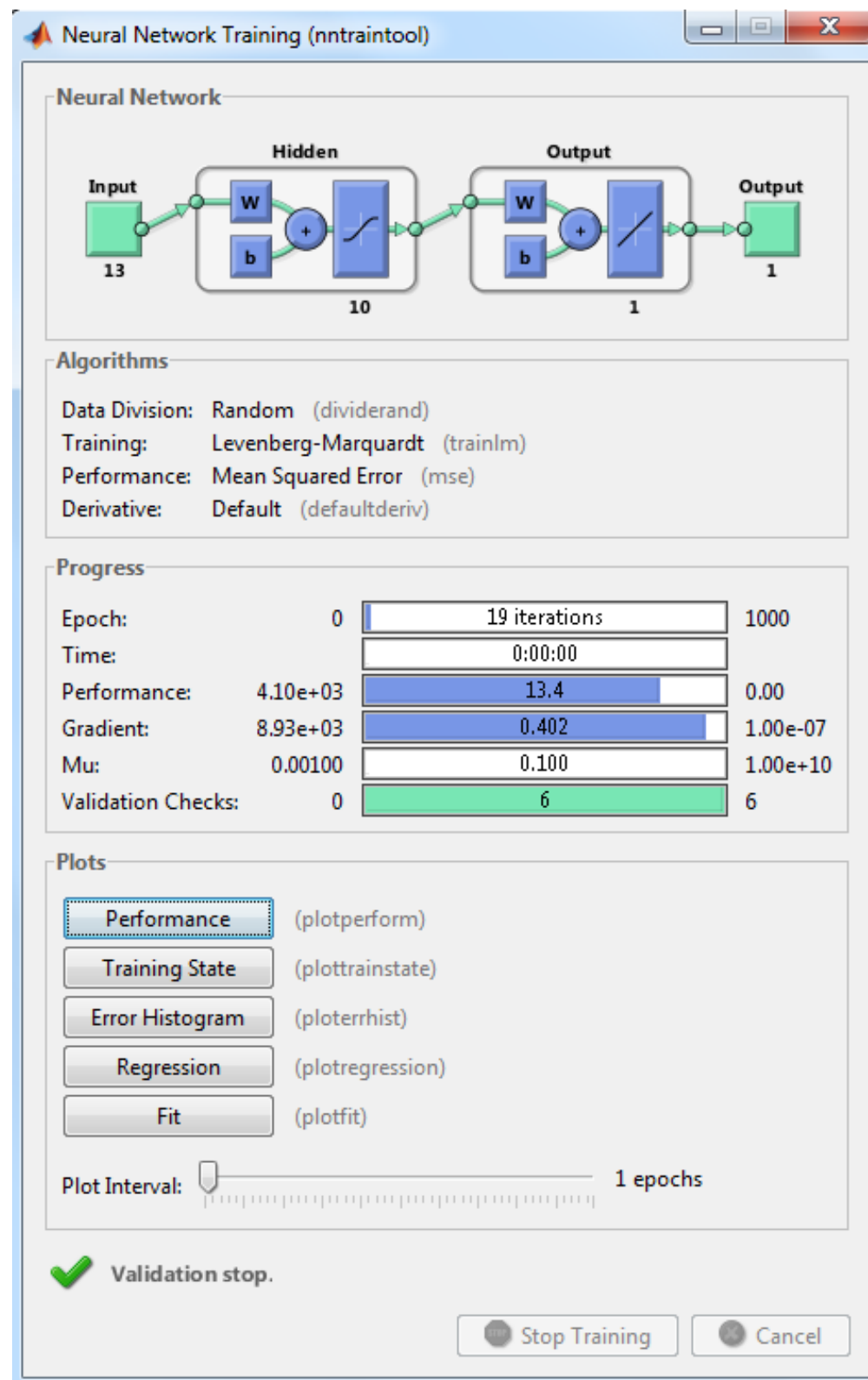
Toolbox de redes neurais do MATLAB



- Toolboxes diferentes para cada tipo de rede neural:
 - Regressor (Fitting)
 - Classificador (Pattern Recognition)
- 2 maneiras de criar e configurar a rede:
 - Uso direto do Toolbox – Mais simples
 - Uso de um script – Mais opções de configuração
- Os nós de cada toolbox possuem funções de transferência fixas

Softwares utilizados

Toolbox de redes neurais do MATLAB



Toolbox de um regressor
(Neural Net Fitting)

- Permite o uso de um conjunto de validação
 - Conjunto de verificação do treino que diminui o N° de épocas necessário
- Permite a realização de rotinas de treino (Loop de execução no script)
- Permite traçar gráficos de desempenho:
 - Histograma de erros
 - Matriz de confusão
 - Curva ROC
 - ...

- Configuração de parâmetros mais ampla:
 - Divisão dos conjuntos de treino, teste e validação
(Randômica, Por índice, ...)
 - Medida principal de qualidade
(MSE, CrossEntropy, ...)
 - Pré-processamento (Normalização, PCA, ...)
 - Algoritmo de processamento
(BackPropagation, Levenberg-Marquardt, ...)
 - ...

- Salvar a rede: objeto “net” no workspace do MATLAB
- O treino por Toolbox tem opções limitadas
- O treino por script exige um conhecimento maior da ferramenta
- Tempo de treino da rede reduzido comparado com o do software Multiple Back-Propagation

Benchmark de CPUs - Regressor de desempenho

Aplicação de RNA em Regressão Linear Multivariada

Apresentação: Pedro Henrique Itio Hayashi

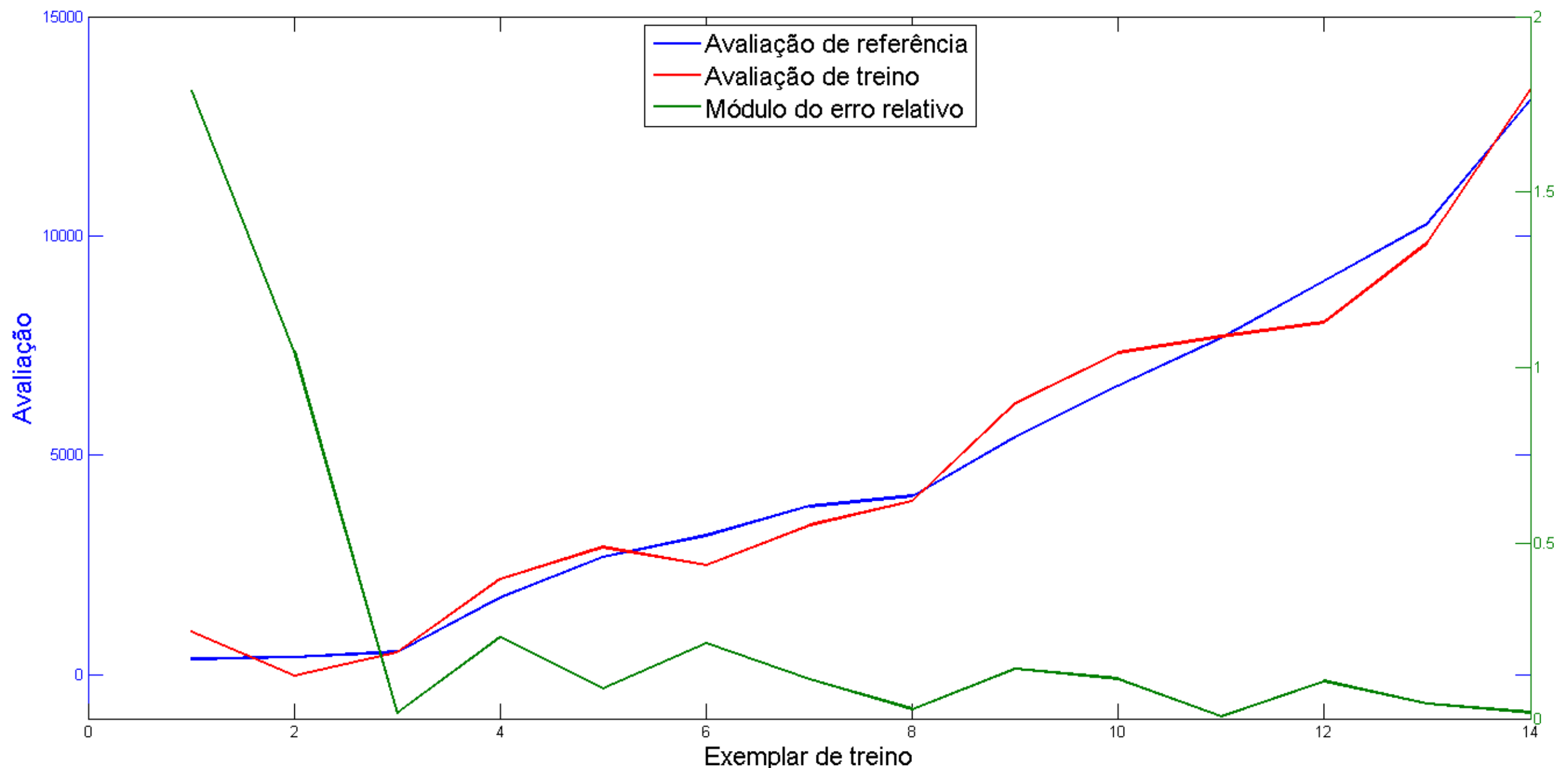
- Tradicionalmente feito através da execução de programas de medição como:
 - Cálculos com inteiros ou ponto flutuante
 - Compressão de dados
 - Busca de números primos
 - Encriptação de dados

- Utilização das características do processador como parâmetros de avaliação:
 - Número de núcleos
 - Clock
 - Memória Cache
 - Tecnologia dos transistores e quantidade
 - Potência

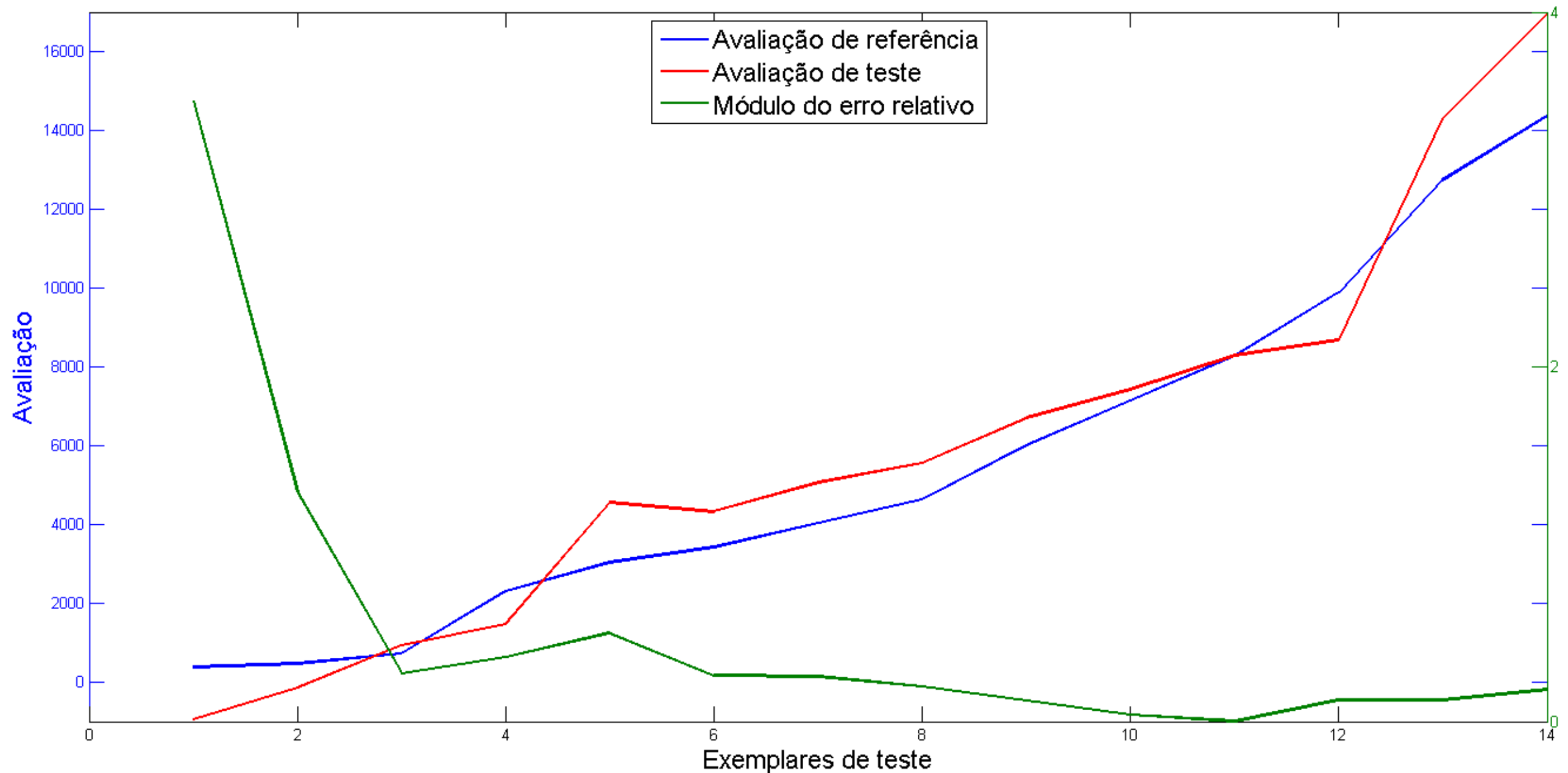
- Extração manual dos parâmetros
- CPUs do ano de 2001 até 2014
- CPUs de uso comum (desktop)
- Uso de 2 fontes de dados:
 - Benchmark: site da PassMark Software
 - Características de CPUs: CPU Database

- Conjuntos reduzidos e fixos para treino e teste
(14 amostras para cada)
- Treino inicial com 1 nó linear
- Medidas de desempenho:
 - RMSE (disponibilizado pelo MBP)
 - Erro relativo = $(\text{saída} - \text{referência}) / \text{referência}$

- Resultados negativos e erros relativos altos
- RMSE treino = 0,0408



- RMSE teste = 0,0914



- Aumento do número de exemplares
 - 100 exemplares para teste e 100 para treino
- Aumento da complexidade da rede
 - Mudança da função dos neurônios
 - Aumento do número de neurônios
- Nova medida de qualidade com base em **[ref]**:
Acerto = módulo do erro relativo $< 10\%$

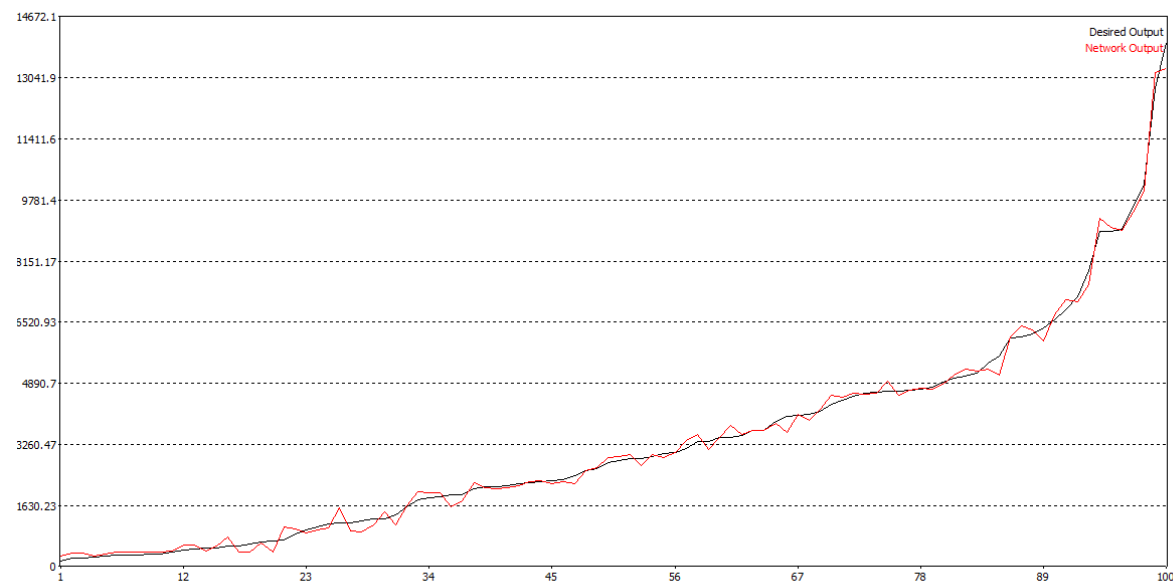
- Treino de redes com topologias mais simples
- Benchmark possui característica não linear
 - Resultados melhores com nós não lineares

Topologia	RMSE Treino	RMSE Teste	Acertos Treino	Acertos Teste
1 nó linear	0,04821	0,05441	35	35
1 nó sigmoide	0,01934	0,01872	43	45
Sigmoide 2x1	0,01099	0,01575	57	50

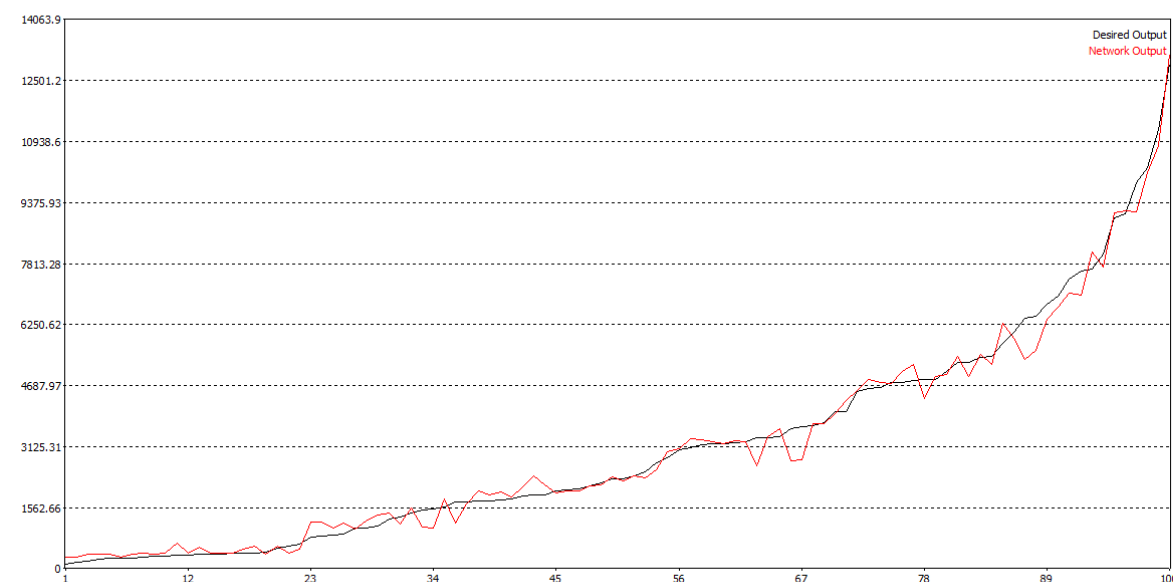
- Avanço para topologias mais complexas
 - Análise do efeito da randomização dos pesos
 - Limitação por efeito de sobreapendizado (overfit)

Topologia	RMSE Treino	RMSE Teste	Acertos Treino	Acertos Teste
Sigmoide 3x1	0,00868	0,01295	67	55
Sigmoide 4x1	0,00904	0,01031	69	61
Sigmoide 5x1	0,00620	0,01169	74	62
5x1 Overfit	0,00571	0,02302	70	50
Sigmoide 6x1	0,00688	0,01097	71	59
6x1 Overfit	0,00378	0,02010	81	52

Rede 6x1 Sigmoidal com treino interrompido

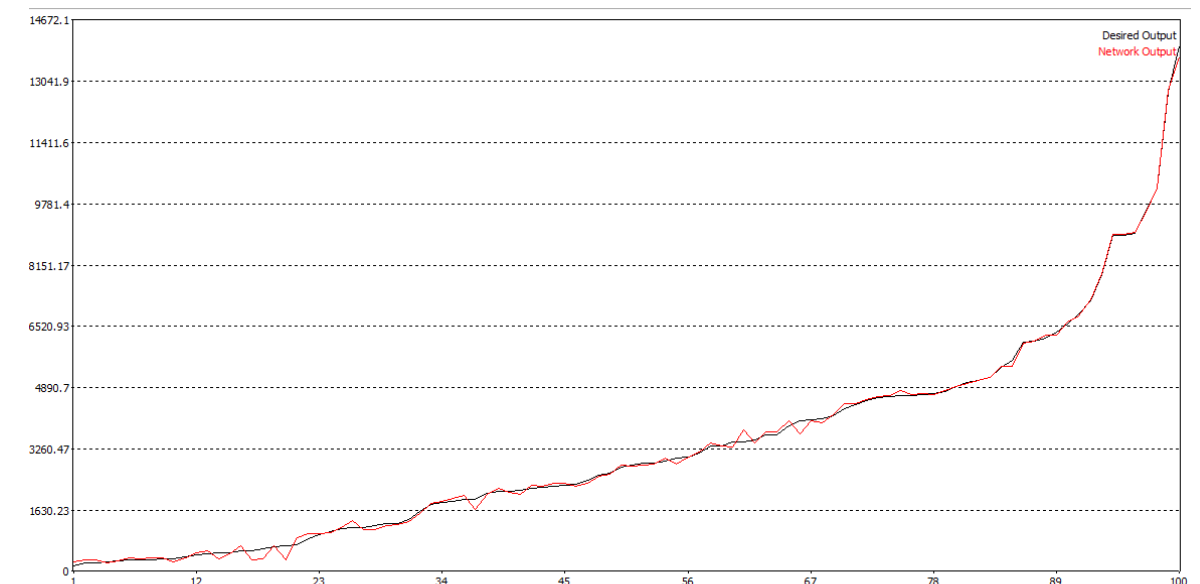


Treino

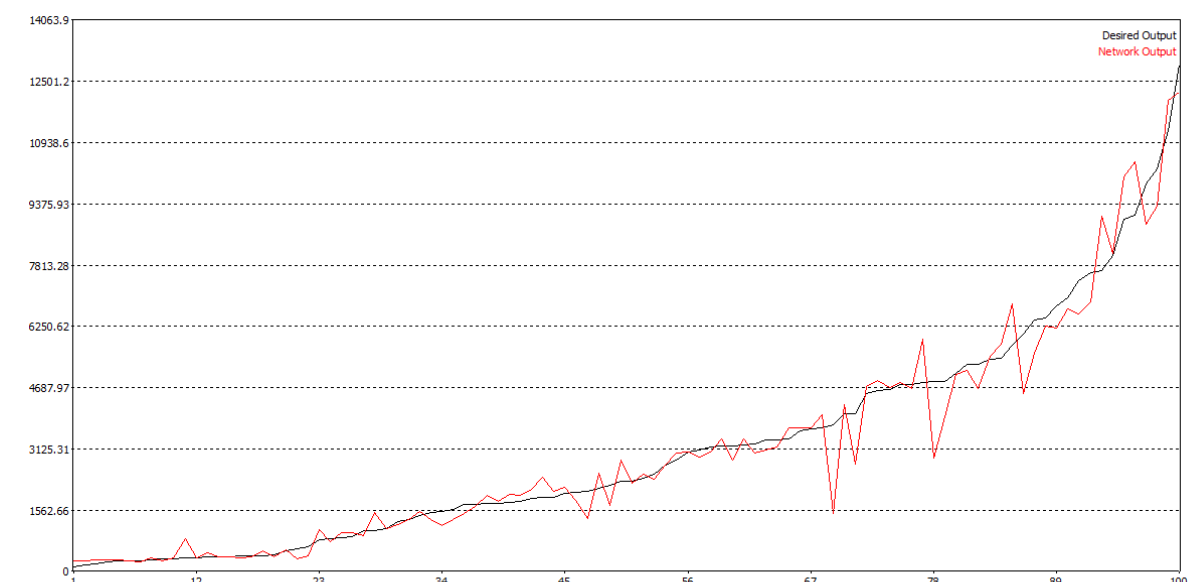


Teste

Rede 6x1 Sigmoidal com overfit



Treino



Teste

- Introdução do conjunto de validação
- Randomização de amostras entre treino e validação de um conjunto de 100
 - 15% Validação, 85% Treino
- Teste fixo
- Medida de qualidade do toolbox: MSE
- Loop de topologias e de randomizações

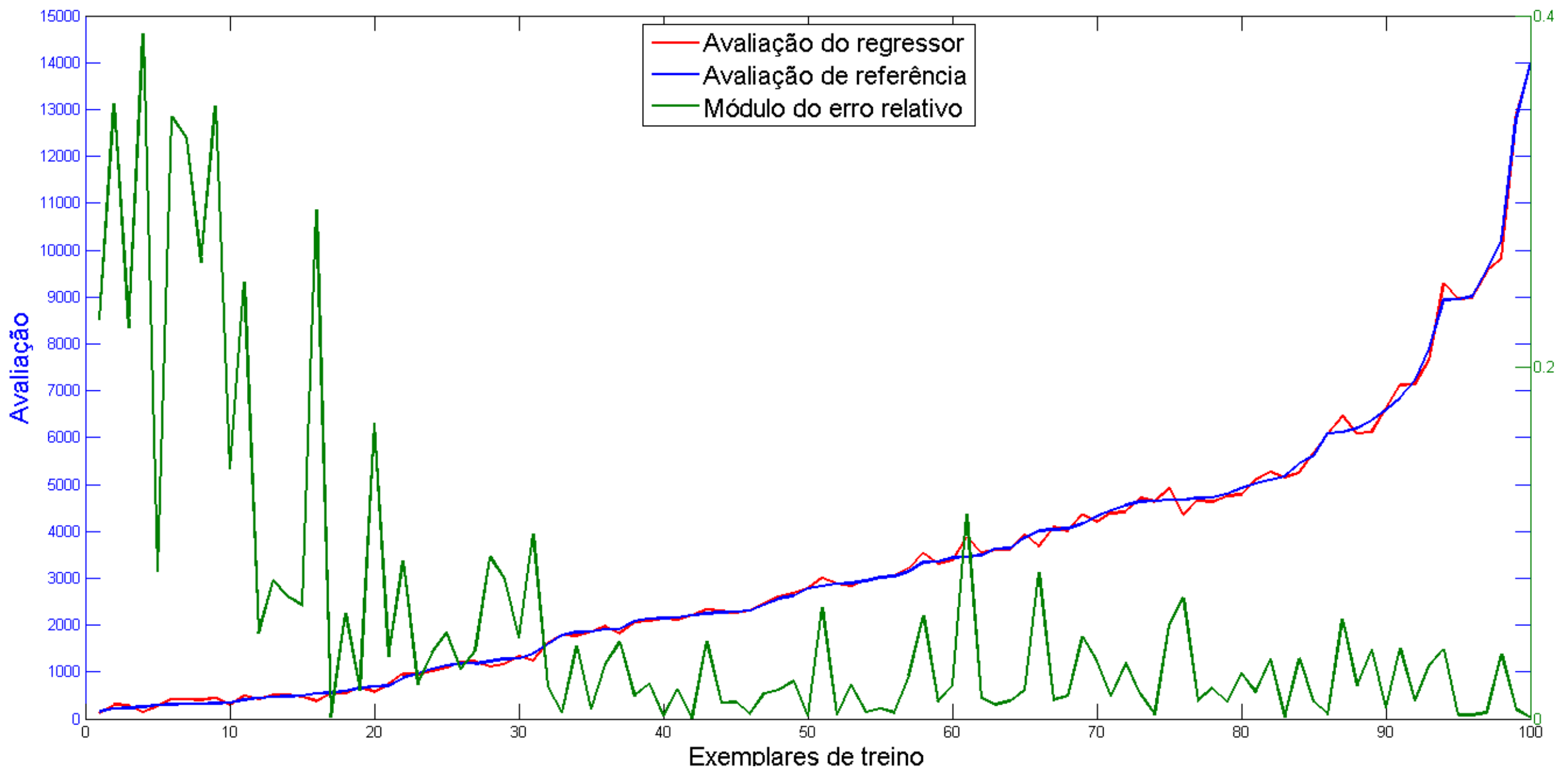
- 1ª simulação
 - 1 a 20 nós , 200 treinos diferentes por topologia
- 2ª simulação
 - 1000 treinos para as topologias com mais acertos
- 3ª simulação
 - 2000 treinos para a rede com maior nº de acertos
 - Conjuntos de treino e teste fixos (sem validação)
 - Escolha da rede neural com melhor resultado

- Vetor de entrada original – **Opção escolhida**
 - 10 características
 - Maior N° de acertos na 1ª e na 2ª simulação
 - Realização da 3ª simulação para fixar rede
- Vetor de entrada reduzido
 - 5 características (características mais comuns)
 - Alternativa como regressor simplificado
- Vetor de entrada com pré-processamento
 - 6 componentes principais (PCA)
 - Menor N° de acertos

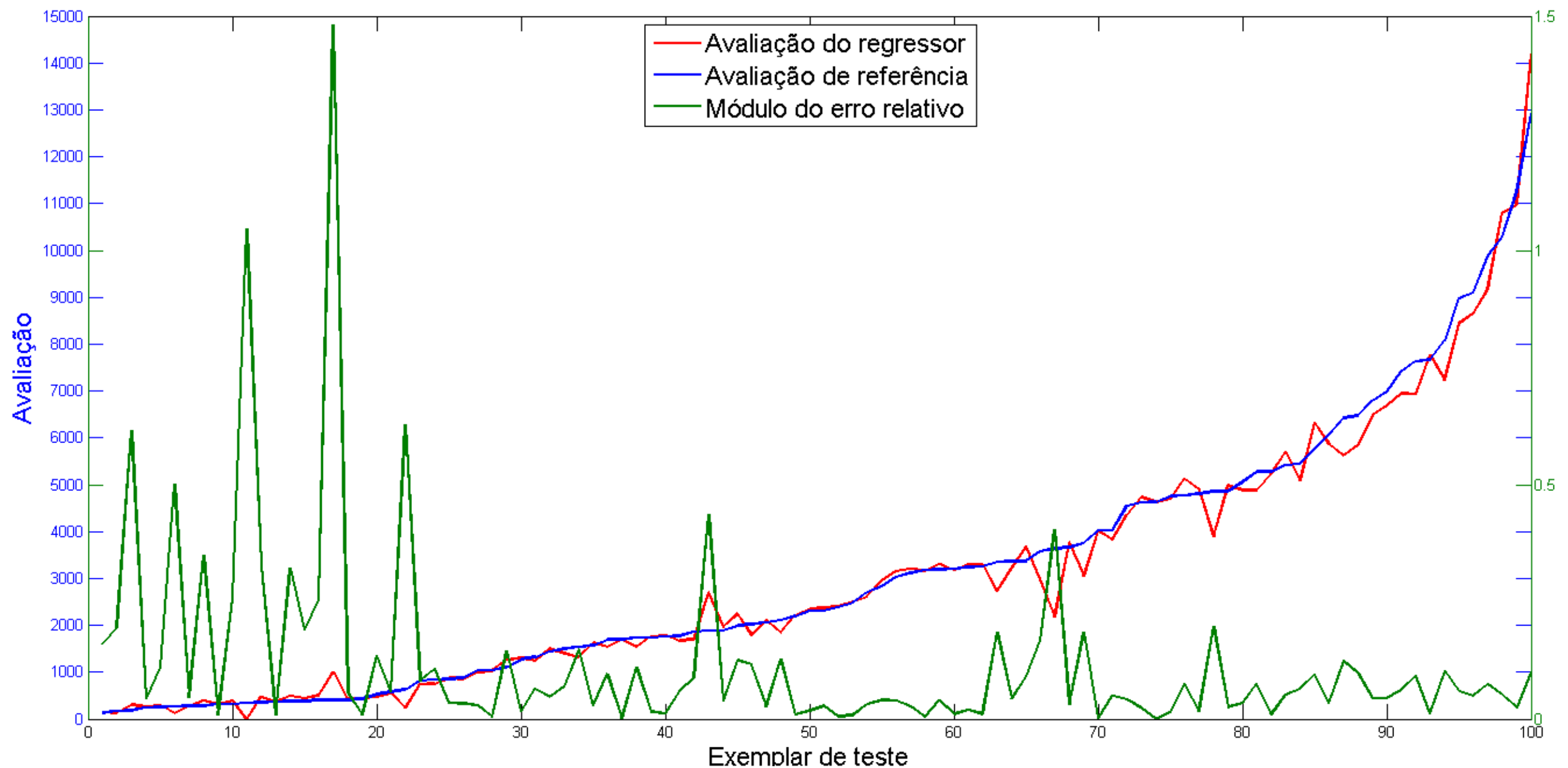
- 15 exemplares de validação
- 85 exemplares de treino
- 100 exemplares de teste
- Média de 1000 treinos

	Nº de nós	MSE Total	Acertos Treino	Acertos Validação	Acertos Teste
10 características	4	815.440	47,5	6,6	43,7
5 características	8	703.104	39,4	5,4	37,0
6 PCA	9	805.507	35,9	4,1	27,4

- 4 nós na camada escondida (Função Sigmoid)
- 86 acertos no treino



- 70 acertos no teste
- Maior erro relativo = 148%



- Redução da entrada piorou o desempenho
 - Entrada de dimensão pequena
 - PCA: componentes de maior contribuição estatística
- Alta variação da saída prejudicou a regressão
 - Na faixa de 100 até aproximadamente 13000
- Erro maior para CPUs de nota baixa (< 1000)
 - Erro grande em comparação com a avaliação
- Melhores resultados para CPUs de nota média-alta

Reconhecedor de caracteres

Aplicação de RNA em Reconhecimento de Padrões

Apresentação: Henrique Seiti Ogawa

- Digitalização de acervos bibliográficos manuscritos e/ou tipografados
- Preservação de conhecimentos técnicos e obras literárias
- Disponibilização e acessibilidade à informação

- Exemplos pré-processados de dígitos (0-9) escritos à mão por mais de 500 pessoas
- Amostra - Matriz 28 x 28, cujos elementos constituem de 0 a 255 (grayscale)
- 60.000 exemplos de treino e 10.000 exemplos de teste validados estatisticamente

- Reconhecedor de dígitos (0 a 9)
- Extração de dados do MNIST usando Matlab
- Remoção de bordas: $28 \times 28 \rightarrow 22 \times 22 = 484$
- Empilhamento das matrizes 22×22 para vetor-linha
- Formato de saída da rede neural: exemplo:

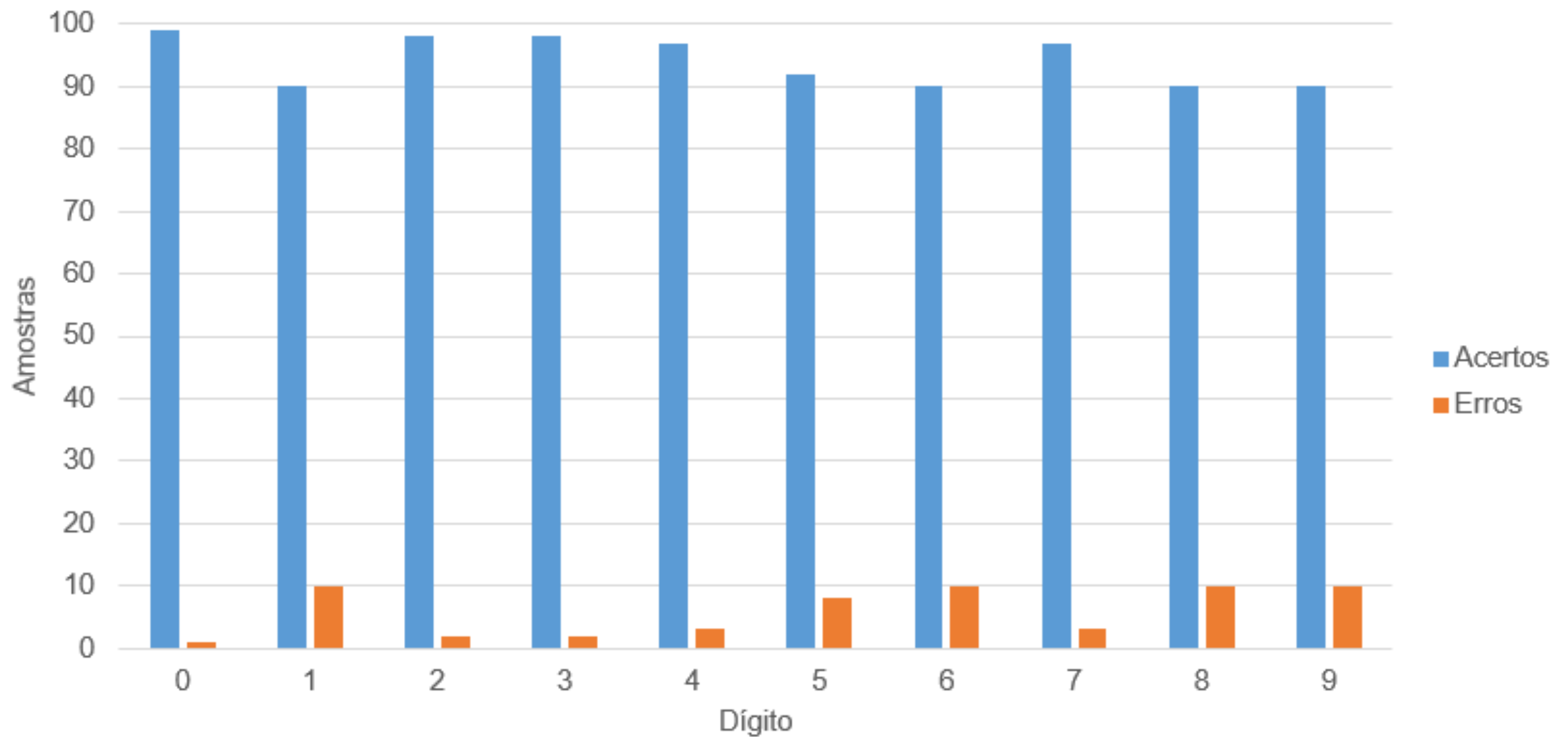
Se dígito 0 $\rightarrow [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$

Se dígito 9 $\rightarrow [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 9]$

- Representação binária dos pixels (fins de classificação)
- Emprego de PCA (Primary Component Analysis)
- Função de transferência tanh nos neurônios
- Treino e teste no MBP (Multiple-Back Propagation)

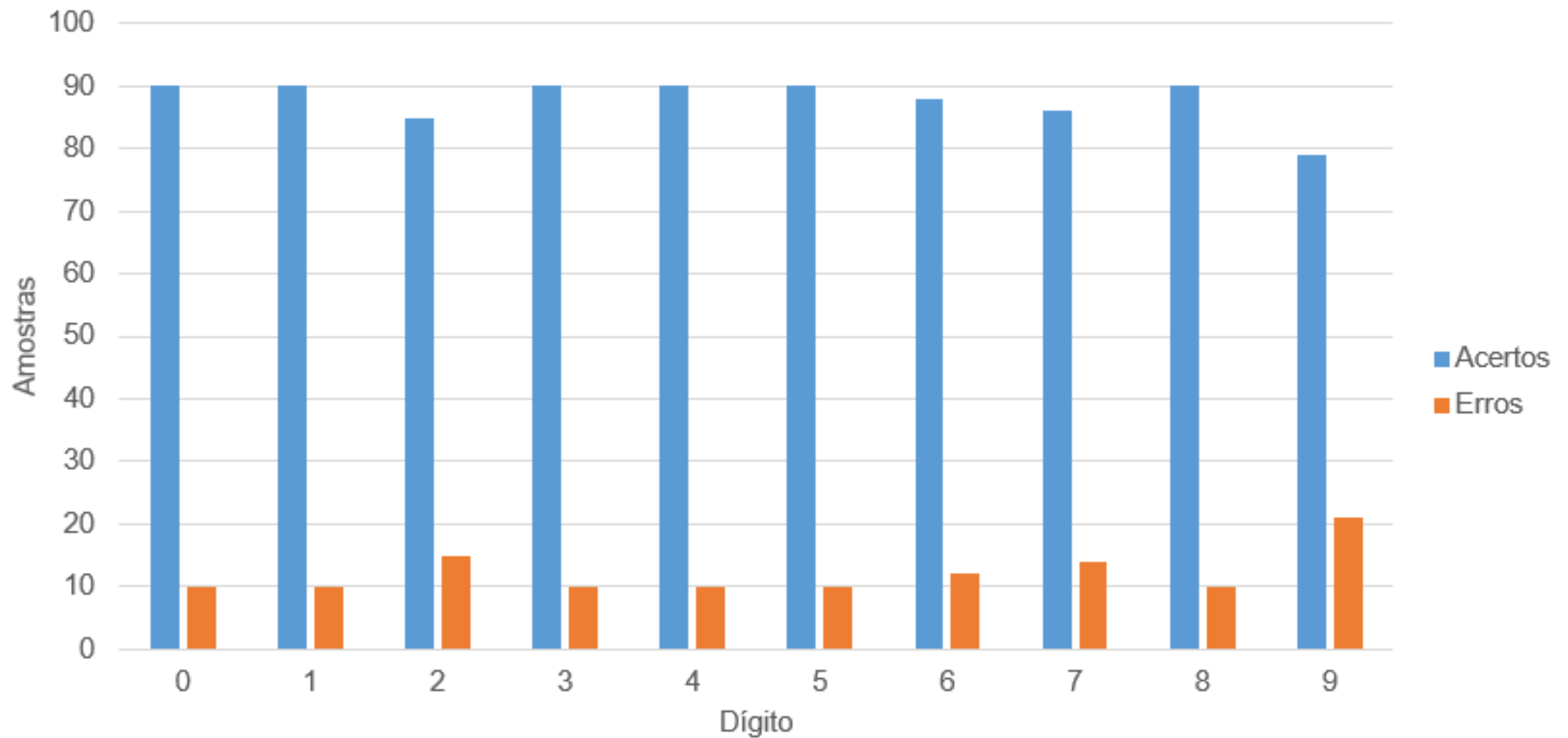
Reconhecimento de Caracteres

Resultados topologia 484-10



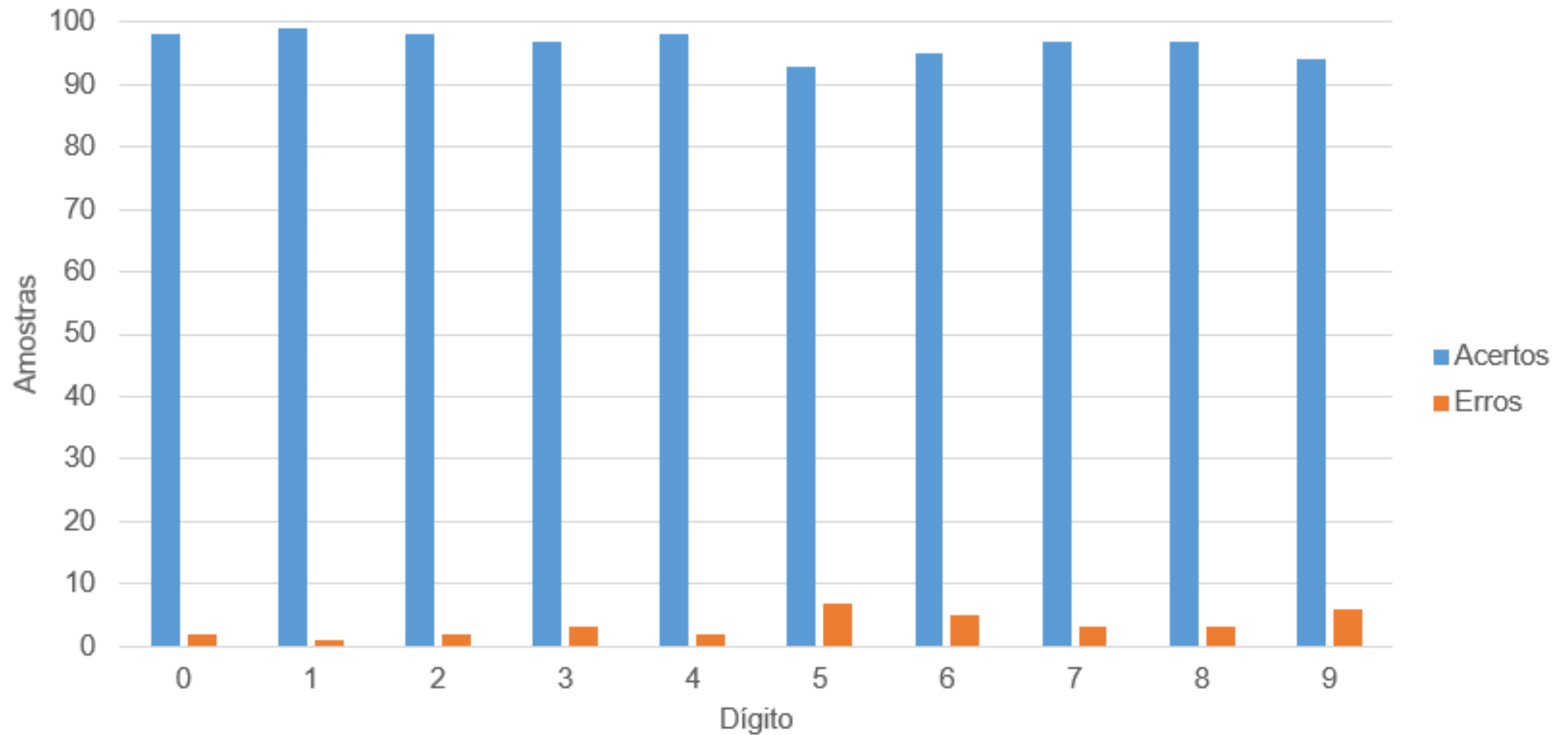
Reconhecimento de Caracteres

Resultados topologia 484-2-10



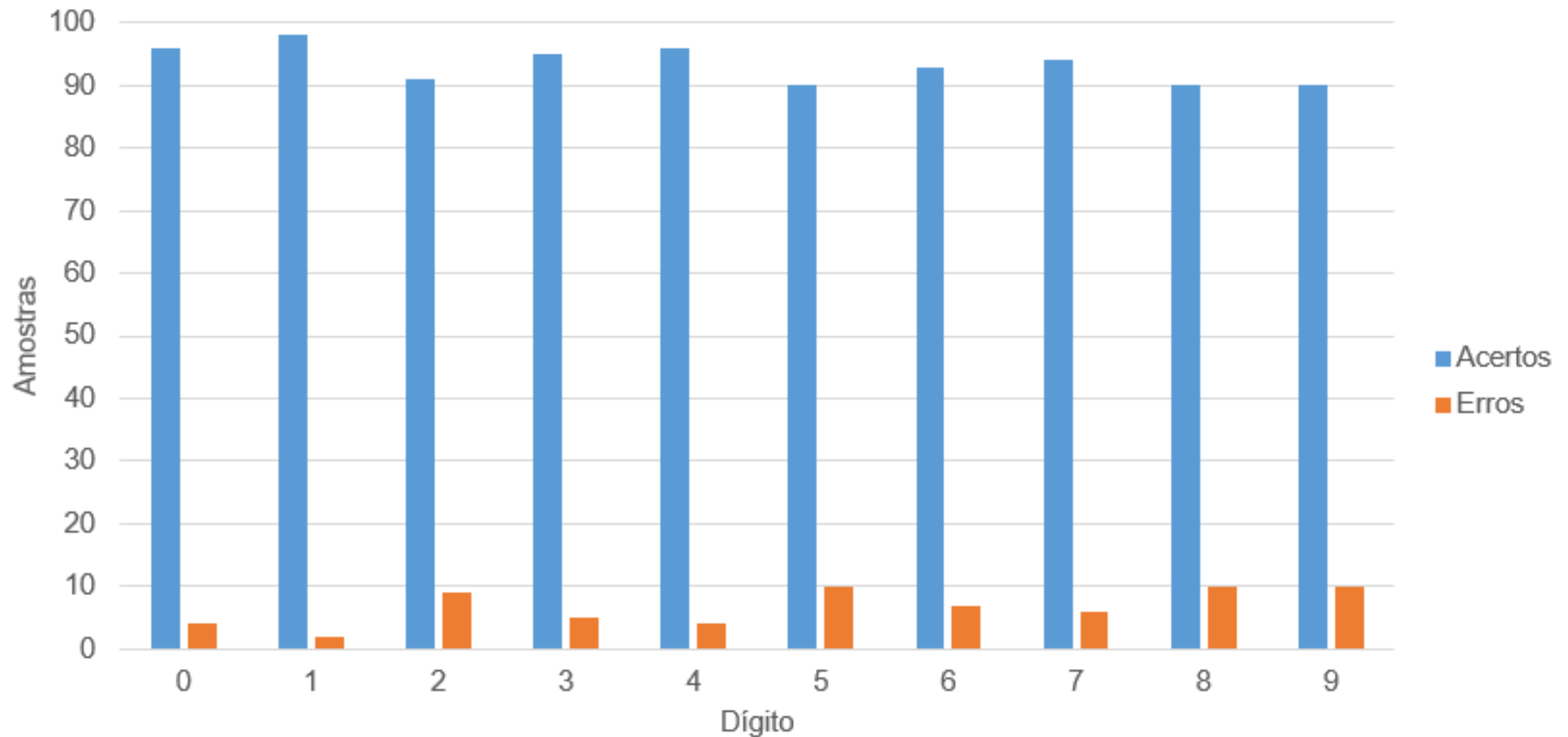
Reconhecimento de Caracteres

Resultados topologia 30-10



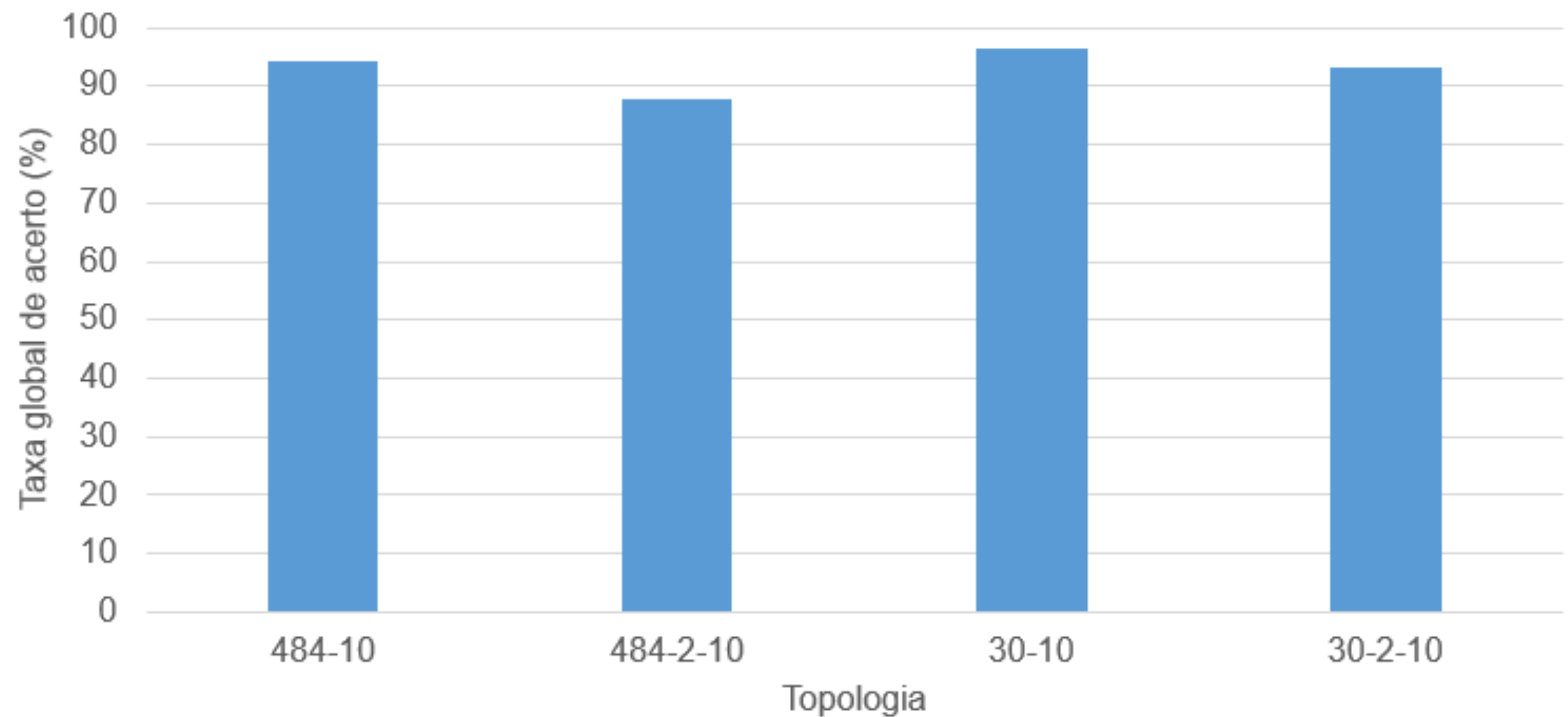
Reconhecimento de Caracteres

Resultados topologia 30-2-10



Reconhecimento de Caracteres

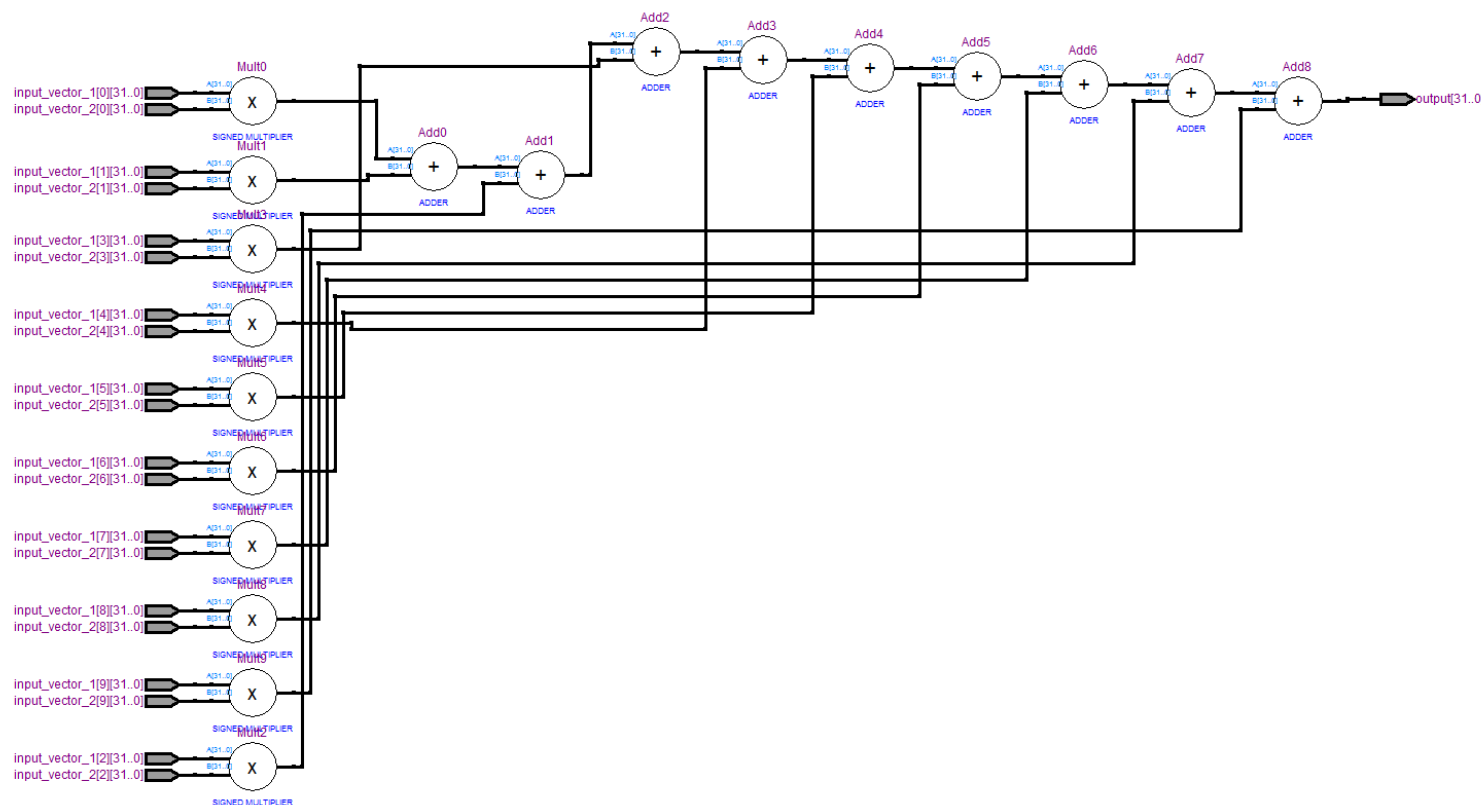
Resultados - topologias



Reconhecimento de Caracteres

Tentativas de embarque

- Tentativa de implementação de uma rede neural MLP 10-10
 - Modelagem de neurônio em hardware digital (FPGA-VHDL)
 - Implementação da operação produto escalar (operação entradas x pesos)
- Primeira implementação do produto escalar: circuito combinatório
- Representação numérica: integer de 32 bits
- Grande utilização de pinos da FPGA

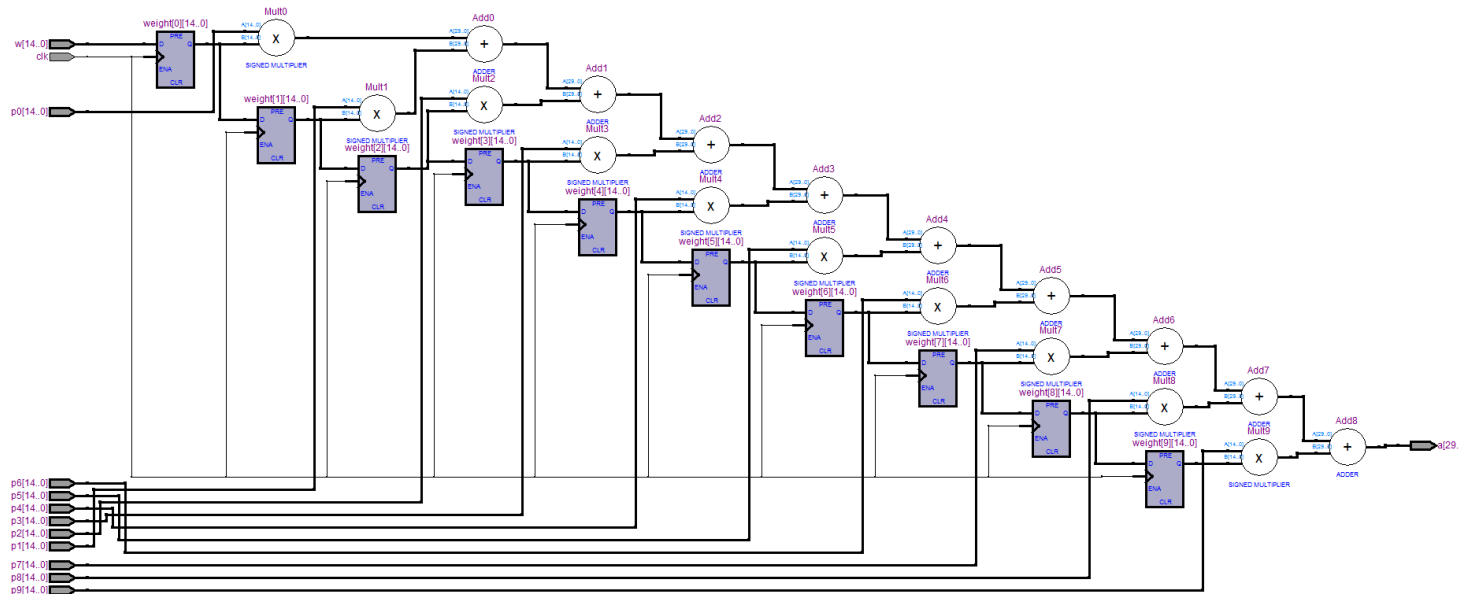


Flow Status	Flow Failed - Fri Jul 03 19:26:26 2015
Quartus II Version	9.1 Build 350 03/24/2010 SP 2 SJ Web Edition
Revision Name	ScalarProduct
Top-level Entity Name	ScalarProduct
Family	Cyclone IV E
Device	EP4CE30F23C7
Timing Models	Preliminary
Met timing requirements	N/A
Total logic elements	569 / 28,848 (2 %)
Total combinational functions	569 / 28,848 (2 %)
Dedicated logic registers	0 / 28,848 (0 %)
Total registers	0
Total pins	672 / 329 (204 %)
Total virtual pins	0
Total memory bits	0 / 608,256 (0 %)
Embedded Multiplier 9-bit elements	60 / 132 (45 %)
Total PLLs	0 / 4 (0 %)

Reconhecimento de Caracteres

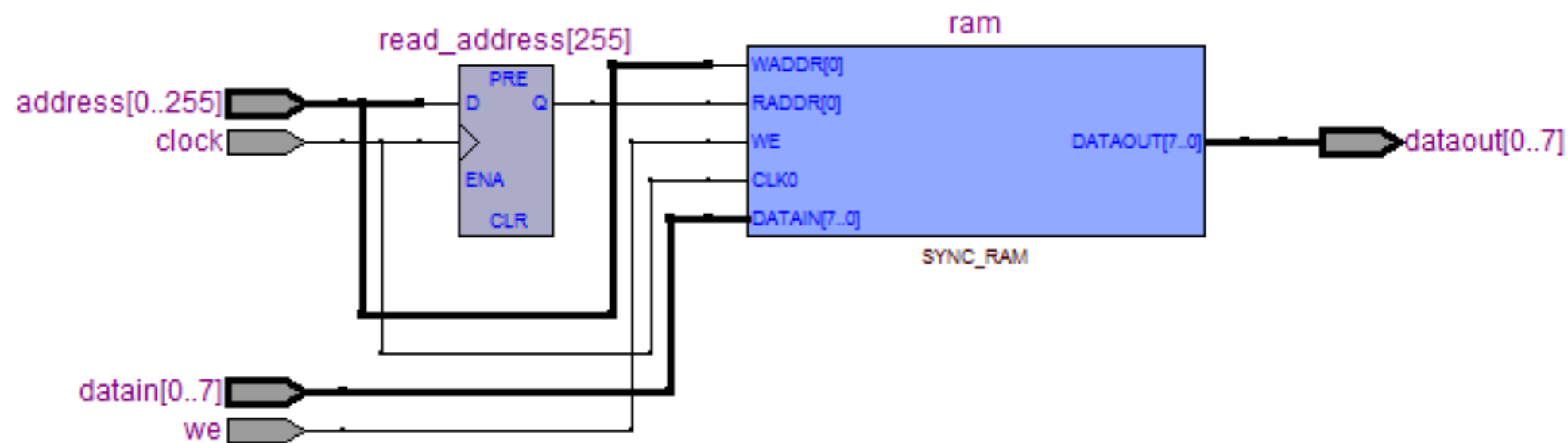
Tentativas de embarque

- Segunda implementação do produto escalar: multiplicador-acumulador
- Representação numérica: integer de 14 bits
- Circuito sequencial, armazenamento de pesos sinápticos em registradores - cálculo sequencial de produtos
- Grande utilização de pinos da FPGA



Flow Status	Successful - Fri Jul 03 19:31:46 2015
Quartus II Version	9.1 Build 350 03/24/2010 SP 2 SJ Web Edition
Revision Name	multacc
Top-level Entity Name	multacc
Family	Cyclone IV E
Device	EP4CE30F23C8
Timing Models	Preliminary
Met timing requirements	N/A
Total logic elements	420 / 28,848 (1 %)
Total combinational functions	270 / 28,848 (< 1 %)
Dedicated logic registers	150 / 28,848 (< 1 %)
Total registers	150
Total pins	196 / 329 (60 %)
Total virtual pins	0
Total memory bits	0 / 608,256 (0 %)
Embedded Multiplier 9-bit elements	20 / 132 (15 %)
Total PLLs	0 / 4 (0 %)

- Terceira implementação do produto escalar: utilização da memória RAM
- Armazenamento de pesos sinápticos e de entradas na memória RAM
- Operações controladas por uma máquina de estados finitos
- Implementação menos paralela, porém requer menor quantidade de recursos da FPGA



Reconhecedor de Acordes

Aplicação de RNA em Reconhecimento de Padrões

Apresentação: Carlos Augusto Bergfeld Mori

- Nem sempre encontra-se a transcrição de uma música
- Dificuldade para iniciantes reconhecerem notas e acordes

Obtenção de acordes



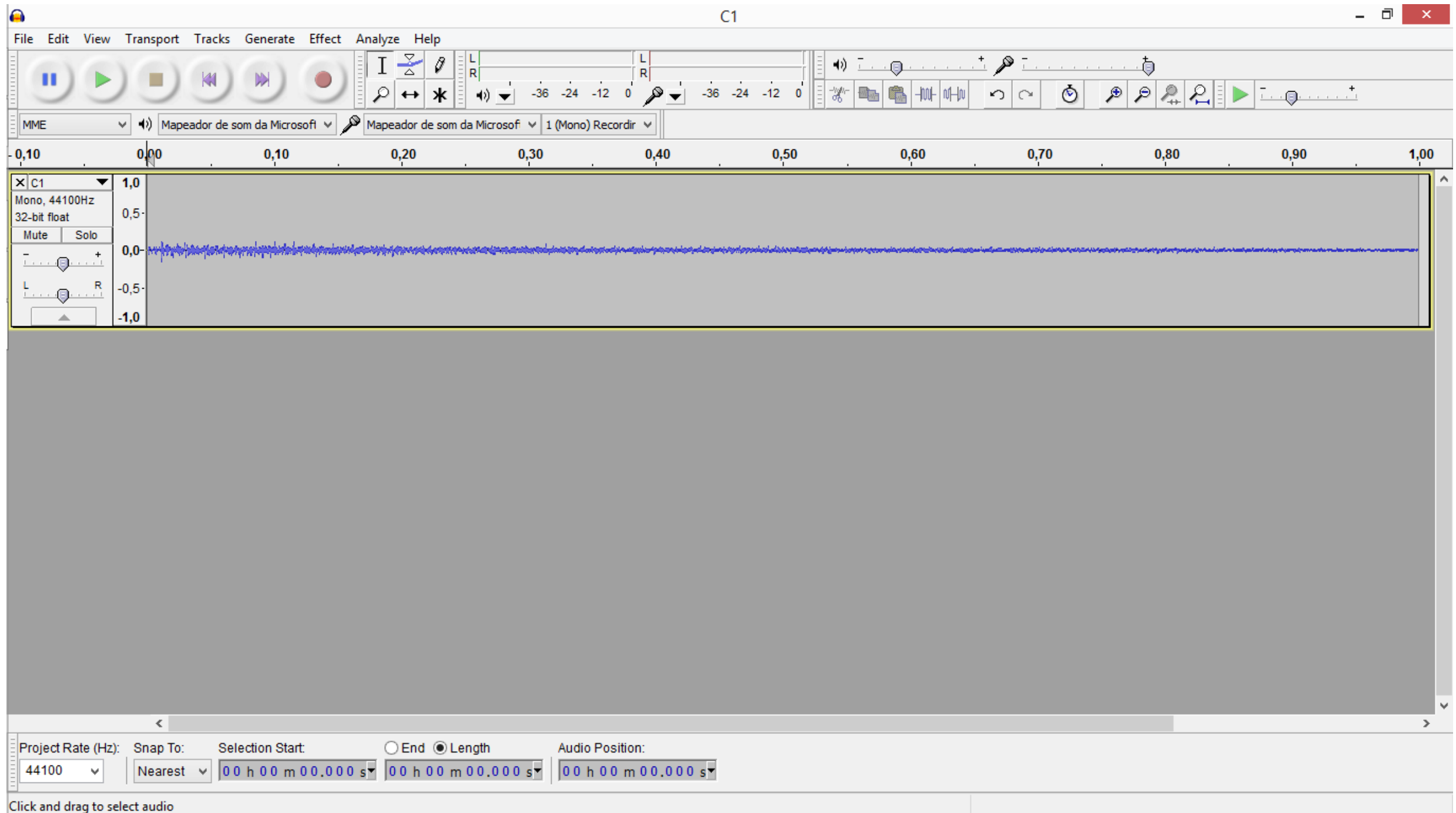
Instrumentos para “ruídos”

University of Iowa
Electronic Music Studios

Fonte: <http://theremin.music.uiowa.edu/>

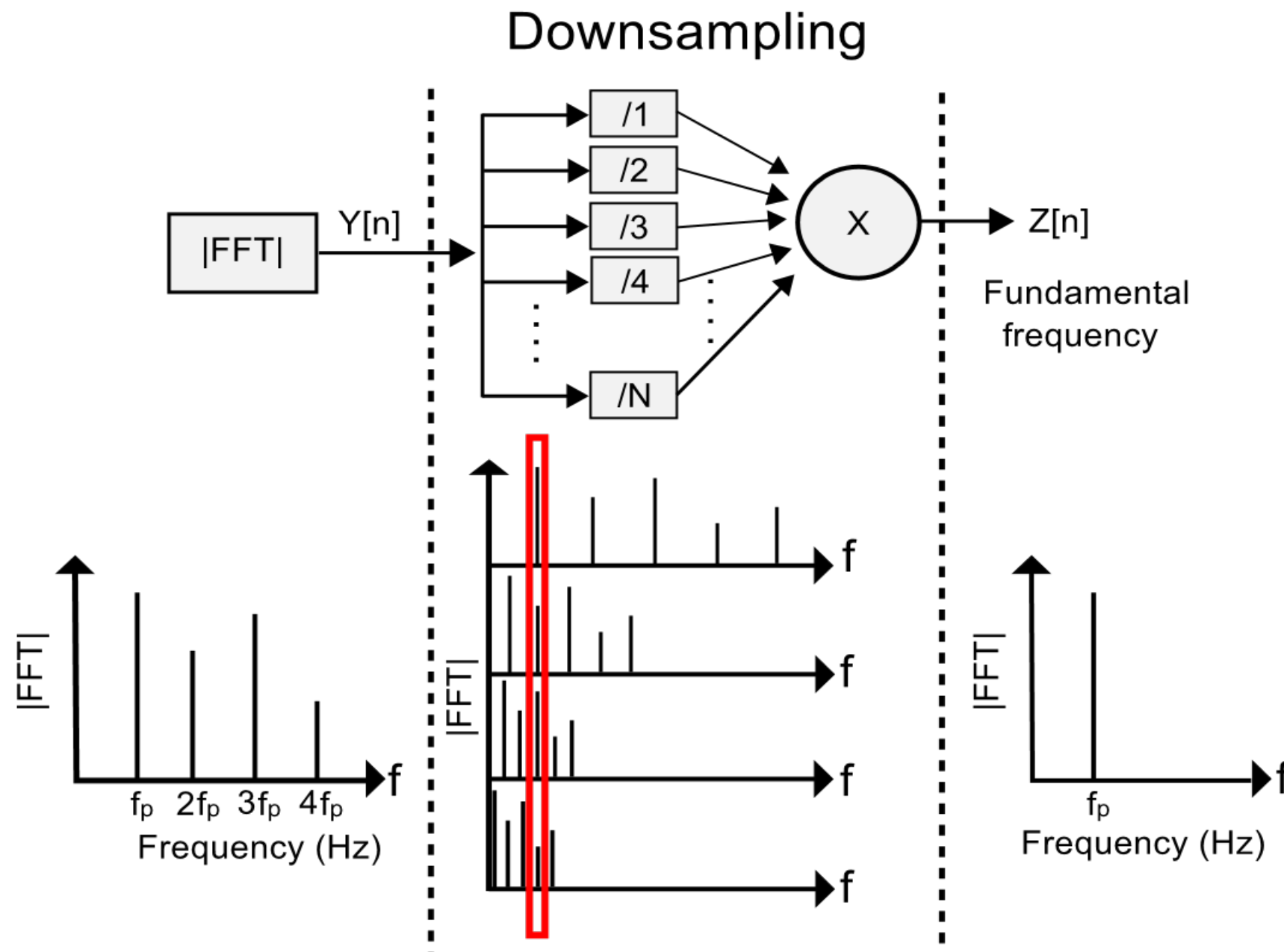
Reconhecimento de Acordes

Obtenção de dados



Reconhecimento de Acordes

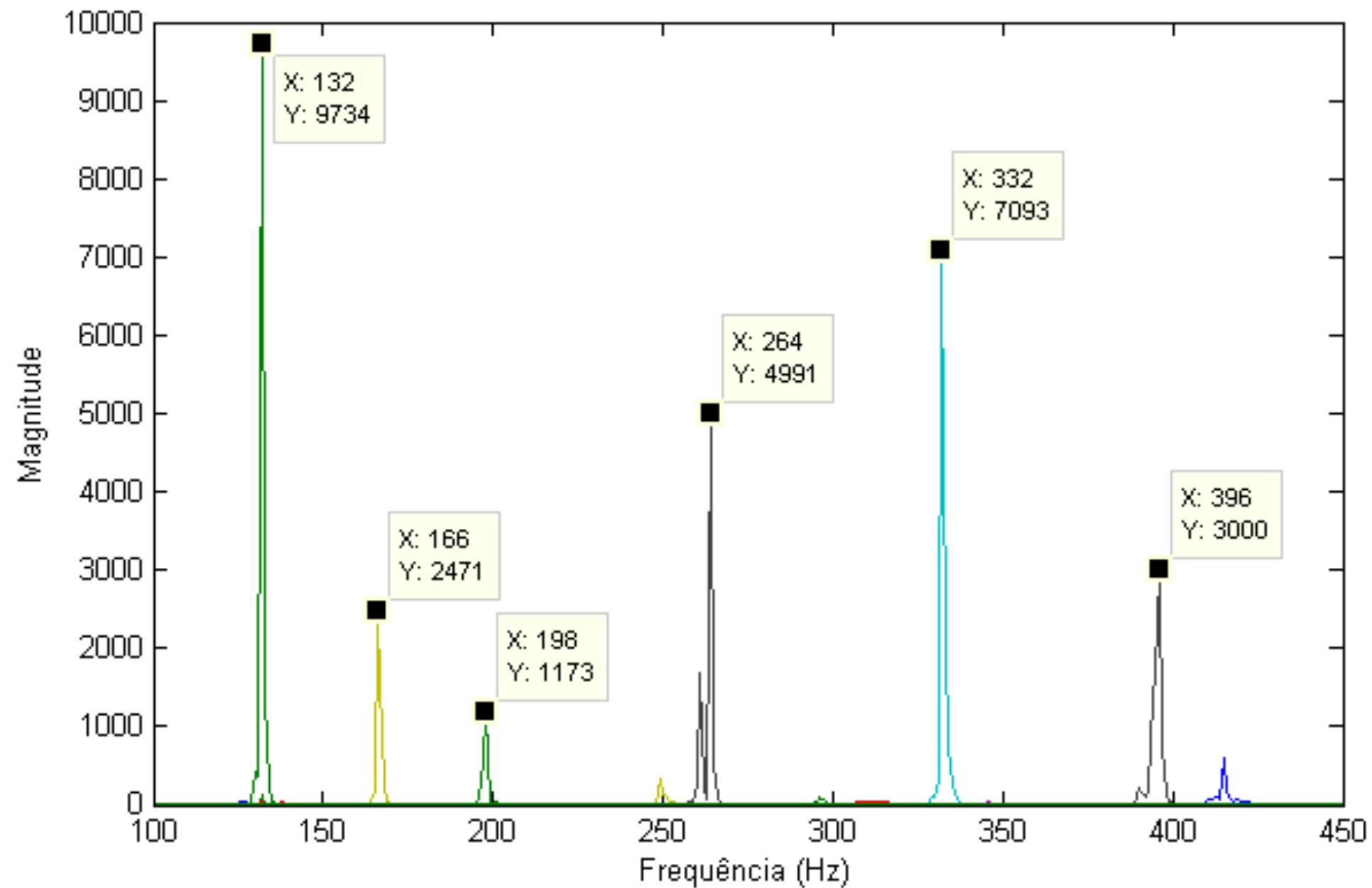
1ª iteração: Pré-processamento



Fonte: J. de Jesus Guerrero-Turrubiates, S. Gonzalez-Reyna, S. Ledesma-Orozco e J. Avina-Cervantes

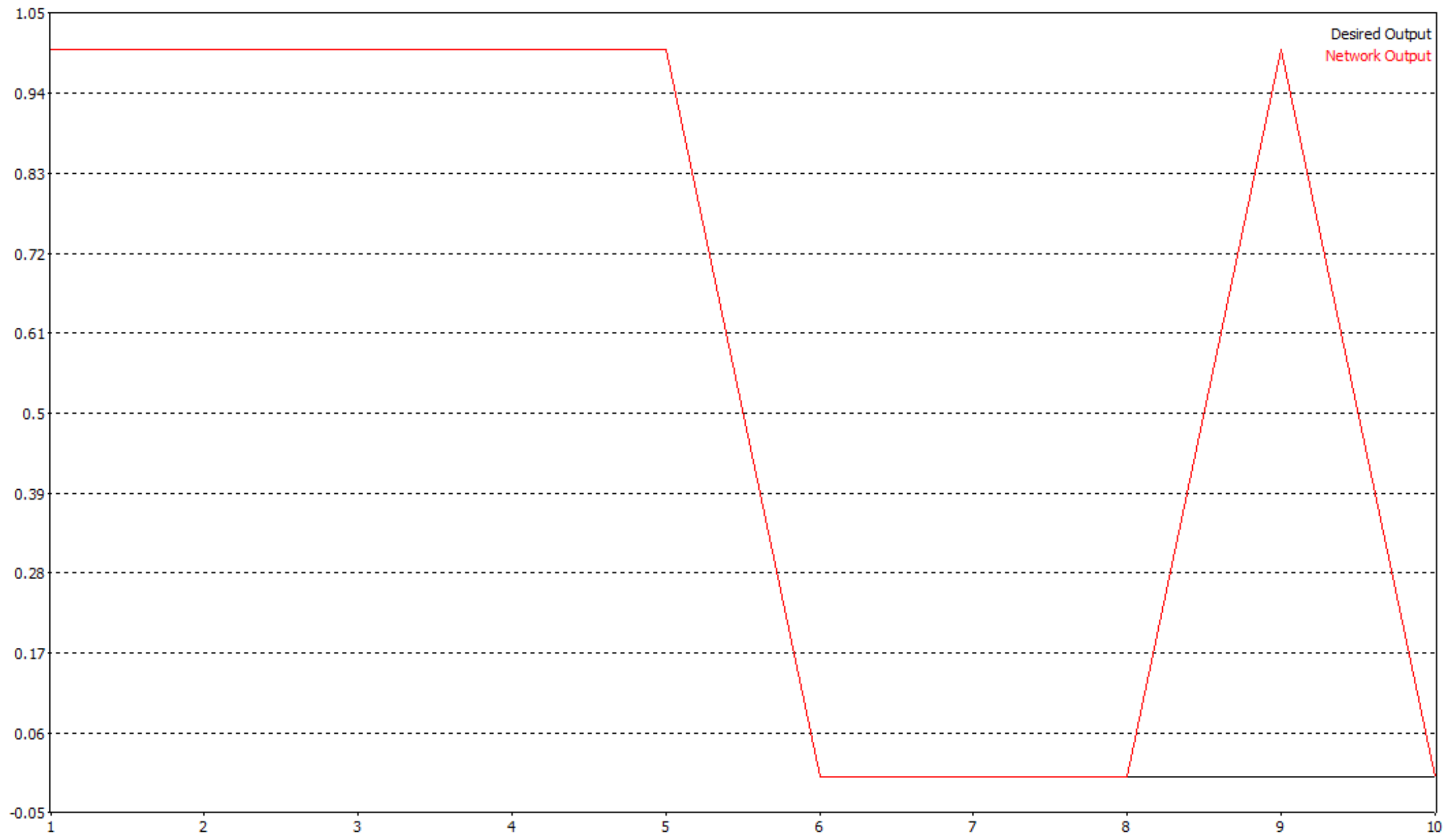
Reconhecimento de Acordes

1ª iteração: Pré-processamento



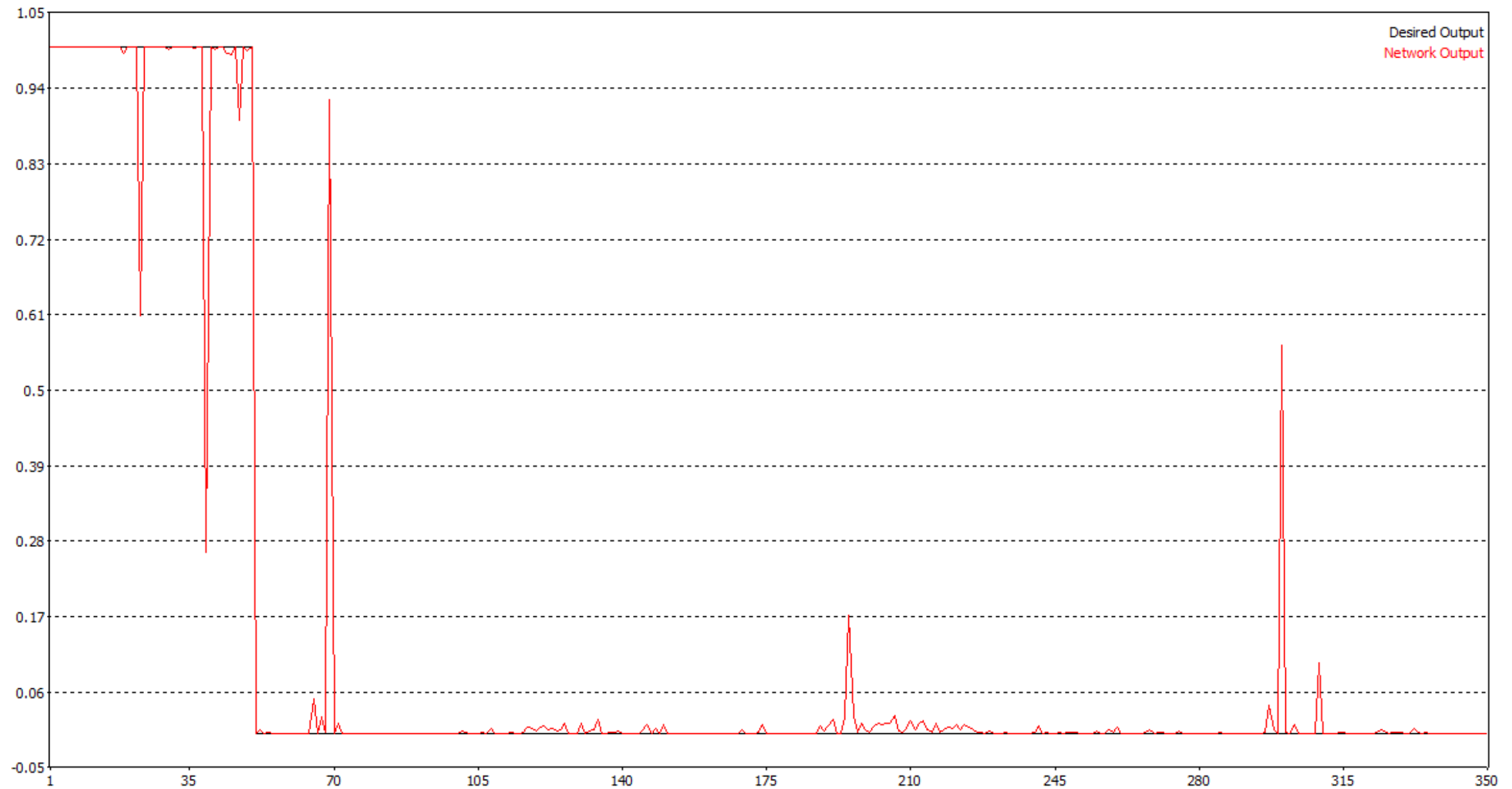
Reconhecimento de Acordes

1ª iteração: Teste com 1 neurônio e 10 dados de 20



Reconhecimento de Acordes

1ª iteração: Teste com 2+1 neurônios e 350 dados de 700



- Modelagem linear
 - Subutilização da RNA
- Não existe diferenciação entre instrumentos
 - Rede capaz de reconhecer apenas acordes isolados

Model-based Approach to Separating Instrumental Music from Single Track Recordings

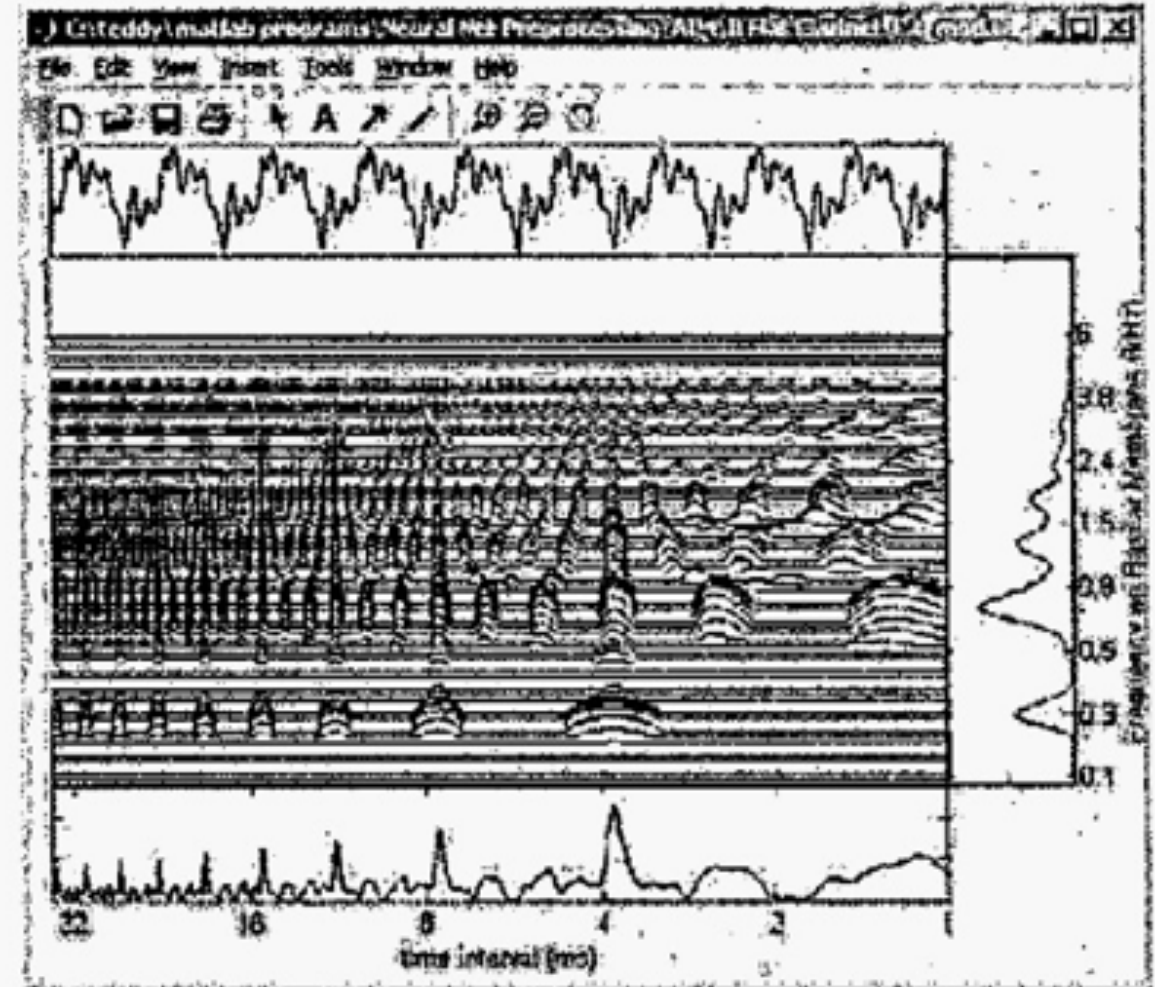
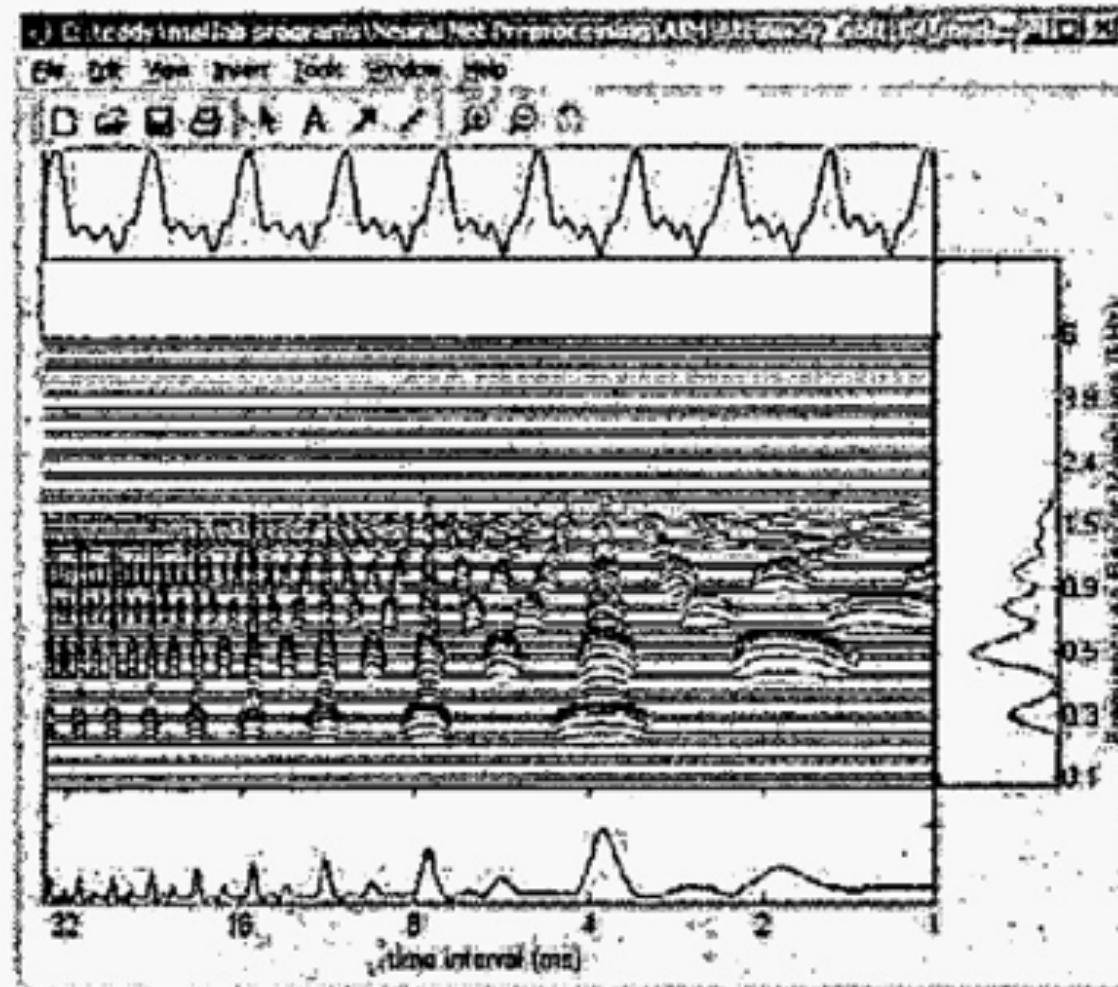
Sintiani D. Teddy and Edmund M-K. Lai



Fonte: <http://www.pdn.cam.ac.uk/groups/cnbh/aimmanual/>

Reconhecimento de Acordes

2ª iteração: Novas possibilidades



Fonte: S. Teddy e E. Lai,

- Partição dos dados:
 - 80% treino - 560 exemplares
 - 10% teste - 70 exemplares
 - 10% validação - 70 exemplares
- Permite traçar gráficos de desempenho:
 - Cada conjunto possui 10 exemplares de cada acorde
 - Ideia: preparar partições para uso do cross-validation

- Loops:
 - Neurônios na camada escondida
 - Randomização dos pesos iniciais
- Critério de ótimo: taxa de acertos no teste
- Loop de cross-validation: sendo implementado

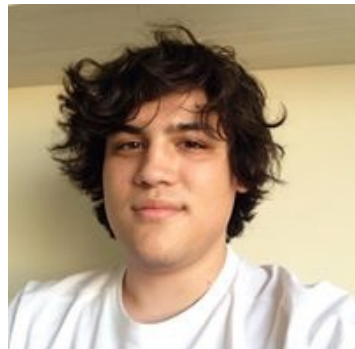
Resultados da rede (teste)

Padrão Ouro

	A	B	C	D	E	F
A	10	0	0	0	0	0
B	0	10	0	0	0	0
C	0	0	10	0	0	0
D	0	0	0	10	0	0
E	0	0	0	0	10	0
F	0	0	0	0	0	10

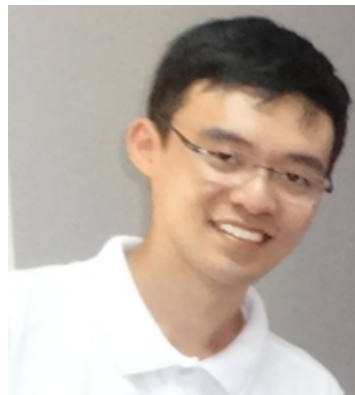
PSI2672 -Apresentação final e demonstração de resultados

Dúvidas? Sugestões?



Carlos Augusto Bergfeld Mori
carlos.mori@usp.br

Reconhecedor de Acordes



Henrique Seiti Ogawa
henrique.ogawa@usp.br

Reconhecedor de Caracteres



Pedro Henrique Itio Hayashi
pedro.hayashi@usp.br

Benchmark de CPUs