



# **Análise Comparativa de Redes Multi – Layer Perceptron e Redes Neuro Fuzzy para Previsão de Vazões médias mensais de Barragens**

**Trabalho de Conclusão de Curso**

**Engenharia da Computação**

**Nome do Aluno: Jheymesson Apolinário Cavalcanti**  
**Orientador: Prof. Mêuser Valença**



UNIVERSIDADE  
DE PERNAMBUCO

**Universidade de Pernambuco  
Escola Politécnica de Pernambuco  
Graduação em Engenharia de Computação**

**Jheymesson Apolinário Cavalcanti**

**Análise Comparativa de Redes Multi –  
Layer Perceptron e Redes Neuro Fuzzy  
para Previsão de Vazões médias  
mensais de Barragens**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

**Recife, dezembro de 2016.**



## MONOGRAFIA DE FINAL DE CURSO

### Avaliação Final (para o presidente da banca)\*

No dia 22 de 12 de 2016, às 10:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente **JHEYMESSON APOLINARIO CAVALCANTI**, orientado pelo professor **Mêuser Jorge Silva Valença**, sob título **Análise Comparativa de Redes Multi-Layer Perceptron e Redes Neuro Fuzzy para Previsões de Vazamento de Barragens**, a banca composta pelos professores:

**Mêuser Jorge Silva Valença**  
**Sérgio Mario Lins Galdino**

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada       Aprovada com Restrições\*       Reprovada

e foi-lhe atribuída nota: 9,0 ( *note* )

\*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 7 dias para entrega da versão final da monografia a contar da data deste documento.

---

*Mêuser Jorge Silva Valença*  
**MÊUSER JORGE SILVA VALENÇA**

---

*Sérgio Mario Lins Galdino*  
**SÉRGIO MARIO LINS GALDINO**

\* Este documento deverá ser encadernado juntamente com a monografia em versão final.

*Dedico este trabalho a todos que,  
de alguma forma, me ajudaram durante esses muitos anos de curso e tornaram o  
curso mais confortável e amigável.*

# Agradecimentos

Agradeço primeiramente a Deus que tem me dado condições de concluir o curso, a Joselma Maria Apolinário Cavalcanti e José de Mello Cavalcanti, meus pais, Jhonisson Apolinário Cavalcanti meu irmão, que de alguma forma contribuíram para a conclusão do curso, a Vanessa Oliveira Ogg minha namorada que tem me dado apoio, a Luiz Augusto Amorin que me acompanhou na maioria dos meus projetos da universidade e ao meu professor Mêuser Valença pela orientação.

# Resumo

Esse trabalho tem por objetivo fazer uma comparação entre a rede Multi layer Perceptron (MLP), que é uma rede clássica para resolução de problemas de previsão, com a rede neuro fuzzy, que apresenta qualidades interessantes para a resolução dos problemas de mesma ordem, porém são redes relativamente novas e por isso não são tão usadas.

Palavras-chave: previsão de vazões, redes neuro fuzzy e *MLP*

# Abstract

This work aims to make a comparison between the Multi-Layer perceptron network (MLP), which is a classic network for solving prediction problems, with the neurofuzzy network, which presents interesting qualities for solving problems of the same order, however are relatively new networks and so are not as widely used

Keywords: river flow forecasts, network neuro fuzzy, *MLP*

# Sumário

<b>Capítulo 1 Introdução</b>	<b>1</b>
1.1 Formulação do problema	2
1.2 Motivação	2
1.3 Estrutura da monografia	3
<b>Capítulo 2 Revisão Bibliográfica</b>	<b>4</b>
2.1 Introdução a Redes MLP	4
2.2 Lógica Difusa (Lógica Fuzzy)	5
2.3 Rede Neuro-Fuzzy	6
2.3.1 Arquitetura ANFIS	7
2.3.1.1 Camada 1	7
2.3.1.2 Camada 2	8
2.3.1.3 Camada 3	8
2.3.1.4 Camada 4	8
2.3.1.5 Camada 5	8
<b>Capítulo 3 Ferramentas</b>	<b>9</b>
3.1 Préprocessamento	9
3.2 MatLab	10
3.2.1 Rede MLP	11
3.2.2 Rede ANFIS	16
3.3 Métricas de performance	19
<b>Capítulo 4 Resultados</b>	<b>20</b>
4.1 Metodologia	20
4.2 Experimentos e Resultados	22

4.2.1 Experimentos realizados na rede MLP	22
4.2.2 Experimentos realizados na rede ANFIS	27
<b>Capítulo 5 Conclusão</b>	<b>32</b>
<b>Bibliografia</b>	<b>34</b>

# Índice de Figuras

<b>Figura 1.</b>	Dados referentes a produção de potência elétrica no Brasil .....	1
<b>Figura 2.</b>	Representação gráfica de um Perceptron .....	5
<b>Figura 3.</b>	Representação gráfica de uma MLP.....	5
<b>Figura 4.</b>	Representação gráfica do paradigma da lógica fuzzy.....	6
<b>Figura 5.</b>	Exemplo de arquitetura anfis.....	7
<b>Figura 6.</b>	Base de dados deslocada .....	10
<b>Figura 7.</b>	Botão de seleção para a pasta da base de dados .....	11
<b>Figura 8.</b>	toolbox do MatLab.....	12
<b>Figura 9.</b>	Janela de configuração da nova rede MLP .....	13
<b>Figura 10.</b>	Janela de treinamento da rede MLP.....	15
<b>Figura 11.</b>	Janela de simulação da rede MLP criada e treinada.....	16
<b>Figura 12.</b>	Acesso ao painel de criação de uma rede ANFIS .....	17
<b>Figura 13.</b>	Painel de criação e uso da rede ANFIS.....	18
<b>Figura 14.</b>	Acesso às regras e à arquitetura.....	19
<b>Figura 15.</b>	Série original das Usinas de Sobradinho e Três Marias, sem normalização dos dados.....	21
<b>Figura 16.</b>	Série da Usina Três Marias, após a normalização.....	21
<b>Figura 17.</b>	Série da Usina Sobradinho, após a normalização.....	22
<b>Figura 18.</b>	Esquema de blocos simplificado da rede MLP.....	22
<b>Figura 19.</b>	Regressão para a usina de Sobradinho.....	23
<b>Figura 20.</b>	Regressão para a usina de Três Marias.....	24
<b>Figura 21.</b>	Melhor validação de performance para Sobradinho.....	25
<b>Figura 22.</b>	Melhor validação de performance para Três Marias.....	26
<b>Figura 23.</b>	Arquitetura ANFIS para Sobradinho.....	28

<b>Figura 24.</b>	Arquitetura ANFIS para Três Marias.....	28
<b>Figura 25.</b>	Curva de erro para os dados de Sobradinho.....	29
<b>Figura 26.</b>	Curva de erro para os dados de Três Marias.....	29
<b>Figura 27.</b>	Resultado da execução da rede ANFIS para Sobradinho.....	30
<b>Figura 28.</b>	Resultado da execução da rede ANFIS para Três Marias.....	30

# Índice de Tabelas

<b>Tabela 1.</b> Resultados dos CC para Sobradinho .....	26
<b>Tabela 2.</b> Resultados dos CC para Três Marias.....	27
<b>Tabela 3.</b> Erro Quadrático Medio para ambas Sobradinho e Três Marias .....	27
<b>Tabela 4.</b> Comparação do EQM entre as redes MLP e ANFIS .....	30
<b>Tabela 5.</b> Desvio padrão para as redes MLP e ANFIS.....	31

# Tabela de Símbolos e Siglas

RNA – Rede Neural Artificial

FIS - Fuzzy Inference System

*MLP- Multi Layer Perceptron*

*ANFIS - Adaptive-Network-Based Fuzzy Inference System*

ANEEL- Agência Nacional de Energia Elétrica

CC – Coeficiente de Correlação

WWF - *World Wildlife Fund*

ONS- Operador Nacional do Sistema Elétrico

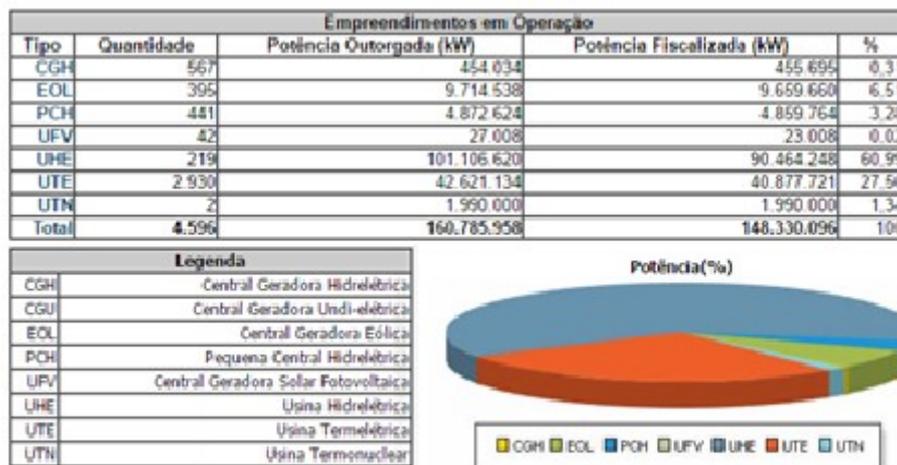
SIN- Sistema Interligado Nacional

# Capítulo 1

## Introdução

A água é o recurso mais abundante no planeta terra, tendo diversas finalidades, sendo que seu excesso ou falta causa problemas de diversas naturezas. Segundo a WWF (*World Wildlife Fund*) [10], com o ritmo atual de consumo de água, até 2025 dois terços da população mundial poderá enfrentar problemas relacionados à escassez de água, por isso o seu gerenciamento é de suma importância e a conscientização e educação da população quanto a isso é um fator imprescindível para o futuro. Por outro lado inundações são responsáveis por 44% das mortes envolvendo desastres naturais em 2013 [11].

No Brasil, a principal fonte de produção de energia elétrica é provinda da água, através das hidrelétricas, mais especificamente mais de 50% da potência total gerada é produzido pelas pequenas centrais hidrelétricas [12] como apresentado na **Figura 1**.



**Figura 1:** Dados referentes a produção de potência elétrica no Brasil

## 1.1 Formulação do problema

Visto a importância que temos diante da produção de energia elétrica no Brasil no tocante à energia provinda das hidrelétricas, sua previsão mediante a histórico de ocorrências de vazões é imprescindível para que haja um melhor planejamento e, conseqüentemente, um melhor gerenciamento dos recursos elétricos produzidos.

O órgão responsável pela previsão de vazões naturais são presente no SIN (Sistema Interligado Nacional) e na ONS (Operador Nacional do Sistema Elétrico)

Os processos e modelos utilizados para esse problema variam segundo a métrica temporal utilizada [13]. Para isso a precisão é um fator imprescindível, porém, apesar de tudo sempre está associada a um fator de erro que depende diretamente da natureza do problema e do horizonte de previsão, que corresponde ao quão distante se pretende prever em relação ao tempo.

## 1.2 Motivação

O planejamento e ações adotados pelo ONS depende diretamente dos resultados esperados para as barragens nas hidrelétricas, e para isso os sistemas que são responsáveis.

Redes Neuro Fuzzy são recomendáveis em aplicações de previsão e classificação por possibilitarem a união de melhorias na precisão do resultado (da lógica fuzzy) com a capacidade de generalização do mesmo (da rede neural) [5] sendo uma rede relativamente pouco usada mas com um potencial enorme para resolver os mesmos problemas.

Com essa premissa, para esse trabalho pretende-se avaliar o desempenho de duas usinas: Sobradinho e Três Marias; comparando seus resultados com a rede clássica usada para resolver problemas de classificação, a rede MLP (*Multi Layer Perceptron*) com a rede Neuro Fuzzy, mais especificamente a rede ANFIS (*Adaptive-Network-Based Fuzzy Inference System*) e validar o erro gerado por ambas as redes.

## 1.3 Estrutura da monografia

No capítulo 2, é apresentada a fundamentação teórica que abrange os principais conceitos sobre redes neurais e matemáticos necessários para a compreensão deste trabalho.

Em seguida o capítulo 3, é apresentada a ferramenta usada para a previsão.

Já no capítulo 4, são descritas as metodologias utilizadas durante o desenvolvimento e os resultados das previsões das redes MLP e ANFIS.

Finalmente, o capítulo 5 contém as conclusões e as sugestões para trabalhos futuros.

# Capítulo 2

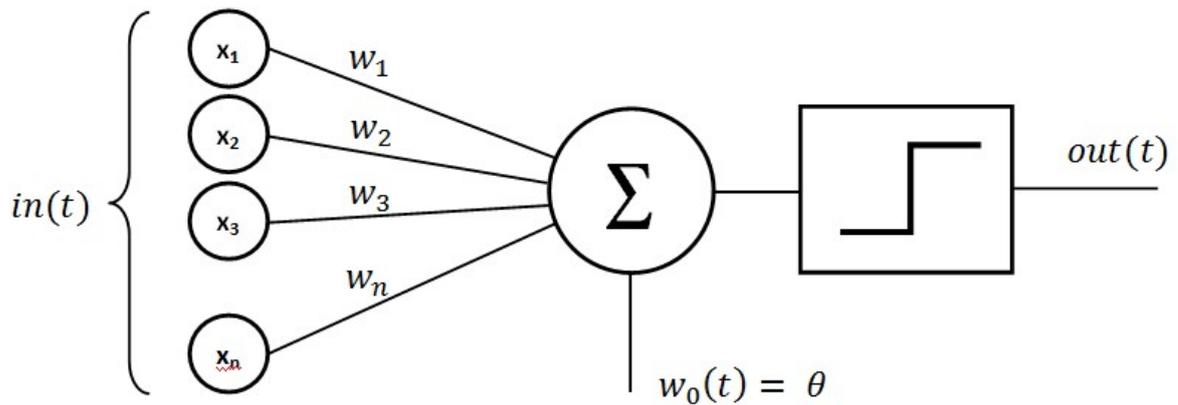
## Fundamentação Teórica

Neste capítulo são apresentados conceitos e definições matemáticas e estatísticas necessárias à compreensão deste trabalho. Na seção 2.1, são apresentadas a noção para redes MLP. Na seção 2.2 são apresentados os conceitos básicos para redes neuro fuzzy. Na seção 2.3 são apresentados os conceitos básicos para as redes neuro fuzzy e as características da arquitetura usada nesse trabalho.

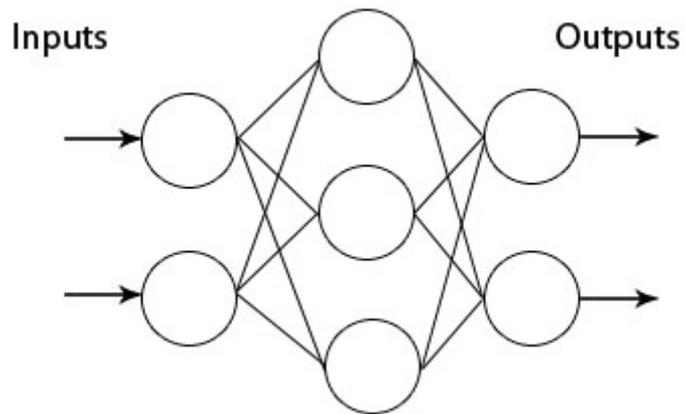
### 2.1 Introdução à redes *MLP*

É uma rede clássica que é composta por vários *perceptrons* com mais de uma camada de neurônios em alimentação direta [1]. O *perceptron* é a rede mais básica criada e apresentada por Rosenblatt em 1958, que consiste em usar um hiperplano para classificar dados linearmente separáveis através de uma função denominada *threshold* como demonstrada na **Figura 2**.

A MLP é ligada através de pesos e sinapses, constituindo camadas sendo que além das camadas de entrada e saída, existe a camada escondida, geralmente existe apenas uma camada escondida como indicado na **Figura 3**, porém a rede não se limita a isso, sendo possível a existência de mais de uma camada escondida implicando diretamente no resultado final. Para esse trabalho nos limitaremos ao uso de uma camada escondida.



**Figura 2.** Representação gráfica de um Perceptron



**Figura 3.** Representação gráfica de uma MLP

Através do tempo as redes MLP têm se mostrado uma excelente ferramenta para classificar e conseqüentemente prever resultados sendo usadas atualmente em diversos sistemas de resposta em tempo real.(por enquanto fim)

## 2.2 Lógica Fuzzy

Diferente da lógica booleana tradicional, que pode ser representado por zero e um, a lógica *fuzzy* é representada por uma escala que vai de zero a um,

representando um grau de pertinência para lidar com conceitos de verdades parciais [4].

Paradoxo de Sorites:

*“Em que momento um monte de areia deixa de sê-lo quando se vai removendo grãos?”*

A lógica difusa lida com classificações como citado acima, onde a transição de um estágio para outro se passa de uma maneira suave, de tal modo que, por muitas vezes, se transita de um estágio a outro sem que se perceba com clareza como apresentado na **Figura 4**; Expressões muito comuns: “alto”, “baixo”, “muito”, “pouco”, “perto”, “longe”, “até o ponto”, etc.



**Figura 4.** Representação gráfica do paradigma da lógica fuzzy

## 2.3 Rede Neuro-Fuzzy

Surgido por volta de 1980 essa técnica se refere a combinação de uma RNA e uma FIS(Fuzzy Inference System), gerando então um novo sistema inteligente híbrido que potencializam as características desses dois paradigmas [2].

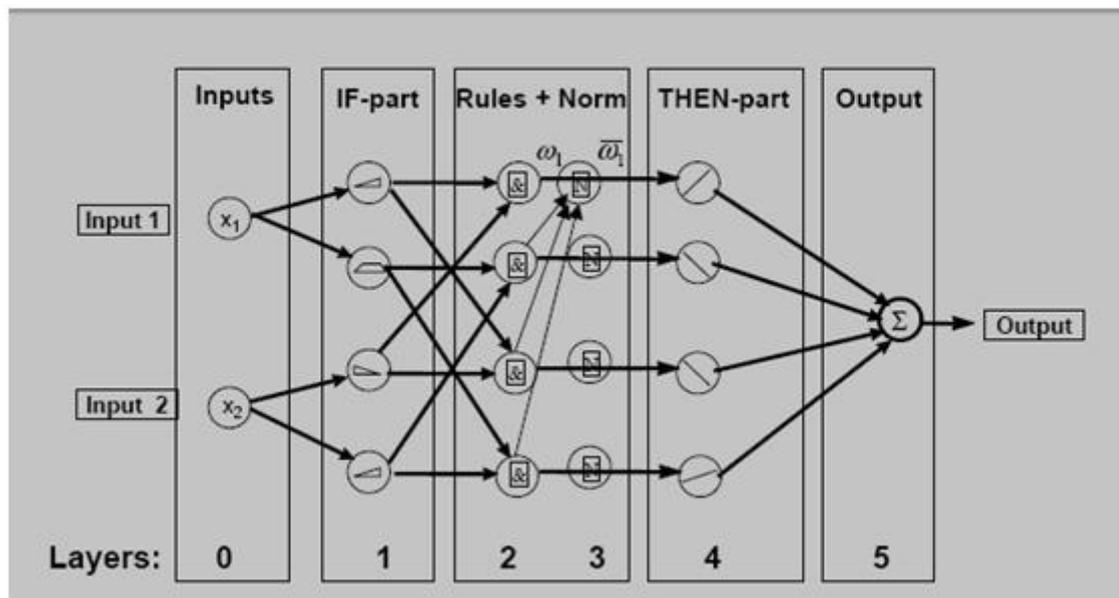
Nesse sistema os termos e regras são aprendidas mediante a apresentação de pares (entrada/saída desejada) apresentando dois comportamentos distintos em fases de aprendizado ou processamento de informação [3].

De uma maneira geral esses sistemas híbridos são redes neurais que utilizam em sua arquitetura a lógica fuzzy, onde são acrescentadas duas novas etapas: a fuzzificação/defuzzificação e um mecanismo de regras [5].

Para esse trabalho utilizaremos a arquitetura ANFIS, cuja a ideia básica é implementar um sistema de inferência fuzzy através de uma arquitetura paralela, no caso a RNA, de forma que os algoritmos de aprendizado possam ser utilizados para ajustar o sistema de inferência [6].

### 2.3.1 Arquitetura ANFIS

O sistema proposto [6] é amplamente usado para resolução de problemas de classificação, e conseqüentemente previsão, sua arquitetura lembra bastante uma rede MLP com varias camadas escondidas, sendo que cada uma das camadas possui uma função específica, como ilustrado na **Figura 5**.



**Figura 5.** Exemplo de arquitetura anfis

**2.3.1.1 Camada 1:** Camada responsável pela etapa de *fuzzificação* do problema, onde os termos lingüísticos são associados e assim cada nó computa o grau de pertinência das entradas. Qual quer função contínua e diferenciável pode ser usada para a ativação dos nós, para este trabalho usou-se uma função gaussiana (Equação 2.1) onde  $a_i$  e  $c_i$  são chamados

de *parâmetros de premissa* que fazem a forma da função variar para se adequar aos conjuntos *fuzzy*.

$$\text{Eq. 2.1 } \mu_{A_i}(x) = \exp \left[ - \left( \frac{x - c_i}{a_i} \right)^2 \right].$$

**2.3.1.2 Camada 2:** Cada nó dessa camada é nomeada por  $\square$ , representa uma regra do conjunto *fuzzy*. Calcula com que grau de pertinência consequente da regra está sendo atendido, ou seja, a força de disparo da regra, para isso ele utiliza um operador de norma T para calcular o  $\omega_i$  (Equação 2.2).

$$\text{Eq. 2.2 } \omega_i = \mu_{A_i}(x) \text{T } \mu_{B_i}, i = 1, 2.$$

**2.3.1.3 Camada 3:** Cada nó dessa camada é chamado de N, e são responsáveis por normalizar os níveis de disparo das regras, calculando o  $\bar{\omega}_i$  (Equação 2.3).

$$\text{Eq. 2.3 } \bar{\omega}_i = \frac{\omega_i}{\omega_1 + \omega_2}, i = 1, 2.$$

**2.3.1.4 Camada 4:** Cada nó desta camada calcula o produto entre os níveis de disparo normalizados e o consequente da regra. A saída dessa camada é o conjunto de parâmetros consequentes (Equação 2.4).

$$\bar{\omega}_i \phi_i = \bar{\omega}_i (p_i x + q_i y + r_i)$$

**2.3.1.5 Camada 5:** Camada de *defuzzificação* e é composta por um único neurônio de ativação linear, que calcula o somatório de todas as entradas ponderadas (Equação 2.5), e consequentemente é a saída do sistema.

$$\phi = \sum_i \bar{\omega}_i \phi_i = \frac{\sum_i \omega_i \phi_i}{\sum_i \omega_i}$$

# Capítulo 3

## Ferramentas

Neste capítulo, é apresentado a principal característica da ferramenta utilizada neste trabalho, para treinamento e previsão de vazões, o *MatLab*. Na seção 3.1 é apresentado o processo de pré-processamento que ocorre na base de dados. Na seção 3.2 são descritos as características principais da ferramenta no *MatLab*. Na seção 3.3 são apresentados as métricas de performance usados para tomar as conclusões desse trabalho.

### 3.1 Pré-processamento

Esse trabalho se atem a duas bases de dados, que foram obtidos através de medidas mensais na unidade de metros cúbicos por segundo de janeiro de 1931 à dezembro de 2003, as bases referidas são: Três Marias e Sobradinho, ambas no rio São Francisco.

Antes de usarmos os dados é necessário que normalizemos os dados, a normalização usada para esse trabalho é a normalização linear (Equação 3.1). A normalização dos dados é necessária para que tenha determinada propriedade [7] e que a rede não priorize os dados numericamente elevados, dessa forma acreditando que os mesmo tem maior impacto no resultado, por isso a normalização é feita e todas as proporções dos dados são respeitadas, a desnormalização não é necessária para esse trabalho, os coeficientes  $a$  e  $b$  podem assumir valores de 0.10 e 0.90 respectivamente ou 0.15 e 0.85 respectivamente, para esse trabalho usaremos a primeira opção.

$$\text{Eq. 3.1 } y = \frac{(a-b)}{(x_{max} - x_{min})} * (x_i - x_{min}) - a, a = 0.10, b = 0.90$$

A próxima etapa é fazer usar a base composta por uma única coluna em várias colunas, para isso o deslocamento da coluna é feita. Para esse trabalho iremos prever apenas um mês a frente. O experimento foi realizado com o

deslocamento temporal de 14 meses atrás, 12 meses, 6 meses e 2 meses, para ambas as bases, o modelo que obtem-se um menor erro foi usando 14 meses, por isso a base foi deslocada temporalmente 14 vezes como indica a **Figura 6**.

										0,102749	0,102834	0,104037	0,1045
									0,102749	0,102834	0,104037	0,1045	0,102611
								0,102749	0,102834	0,104037	0,1045	0,102611	0,101247
							0,102749	0,102834	0,104037	0,1045	0,102611	0,101247	0,100899
						0,102749	0,102834	0,104037	0,1045	0,102611	0,101247	0,100899	0,100734
					0,102749	0,102834	0,104037	0,1045	0,102611	0,101247	0,100899	0,100734	0,100621
				0,102749	0,102834	0,104037	0,1045	0,102611	0,101247	0,100899	0,100734	0,100621	0,100574
			0,102749	0,102834	0,104037	0,1045	0,102611	0,101247	0,100899	0,100734	0,100621	0,100574	0,100942
		0,102749	0,102834	0,104037	0,1045	0,102611	0,101247	0,100899	0,100734	0,100621	0,100574	0,100942	0,101121
0,102749	0,102834	0,104037	0,1045	0,102611	0,101247	0,100899	0,100734	0,100621	0,100574	0,100942	0,101121	0,102327	0,102327
0,102834	0,104037	0,1045	0,102611	0,101247	0,100899	0,100734	0,100621	0,100574	0,100942	0,101121	0,102327	0,102653	0,102653
0,104037	0,1045	0,102611	0,101247	0,100899	0,100734	0,100621	0,100574	0,100942	0,101121	0,102327	0,102653	0,101785	0,101785
0,1045	0,102611	0,101247	0,100899	0,100734	0,100621	0,100574	0,100942	0,101121	0,102327	0,102653	0,101785	0,100969	0,100969
0,102611	0,101247	0,100899	0,100734	0,100621	0,100574	0,100942	0,101121	0,102327	0,102653	0,101785	0,100969	0,100606	0,100606

**Figura 6.** Base de dados deslocada

Para o algoritmo de treinamento usado, a *back-propagation*, é necessário a divisão da base em 3, 50% para treino, 25% para validação e 25% para teste.

No específico a rede MLP, foram feitos os testes variando a quantidade de neurônios na camada escondida, para isso foram feitos experimentos com 14, 20 e 30 neurônios, sendo que com 20 neurônios a curva de calibração teve um melhor coeficiente de correlação.

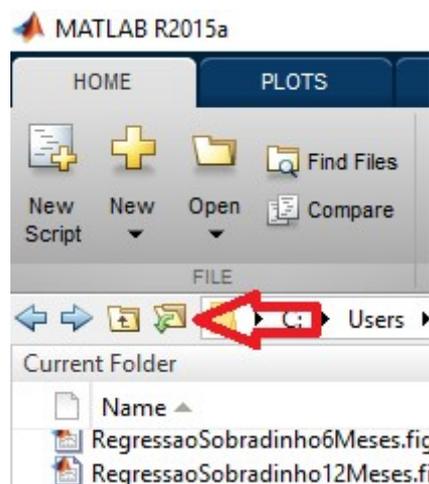
## 3.2 Matlab

O *Matlab* é uma ferramenta largamente usada no ramo da engenharia, dependendo da aplicação o uso demanda muita memória do computador, sua estrutura interna está programada em Java, e possui bibliotecas úteis para esse trabalho, o qual usaremos redes MLP e a ANFIS, a versão do *Matlab* utilizada é a *R2015a*.

### 3.2.1 Redes MLP

De início é necessário importar a base de dados selecionando a pasta que está a base de dados desejada no botão indicado na **Figura 7** e com o comando *load* a base de dados é carregada no *workspace*, e a base de dados deve ser

manipulada para o uso correto da ferramenta, o formato da base usada é: 14 colunas para indicar os meses de entrada e 1 coluna para indicar a saída; a rede gerada pela toolbox exige que as entradas e saídas sejam separadas em arquivos diferentes e, tanto a entrada como a saída, devem estar nas linhas, para esse caso usou-se uma variável criada no momento e ela recebeu o comando *transpose* com a base de dados de entrada e ou saída para modelar corretamente a base de dados e todas as entradas e saídas ficaram nas linhas, como a ferramenta exige.

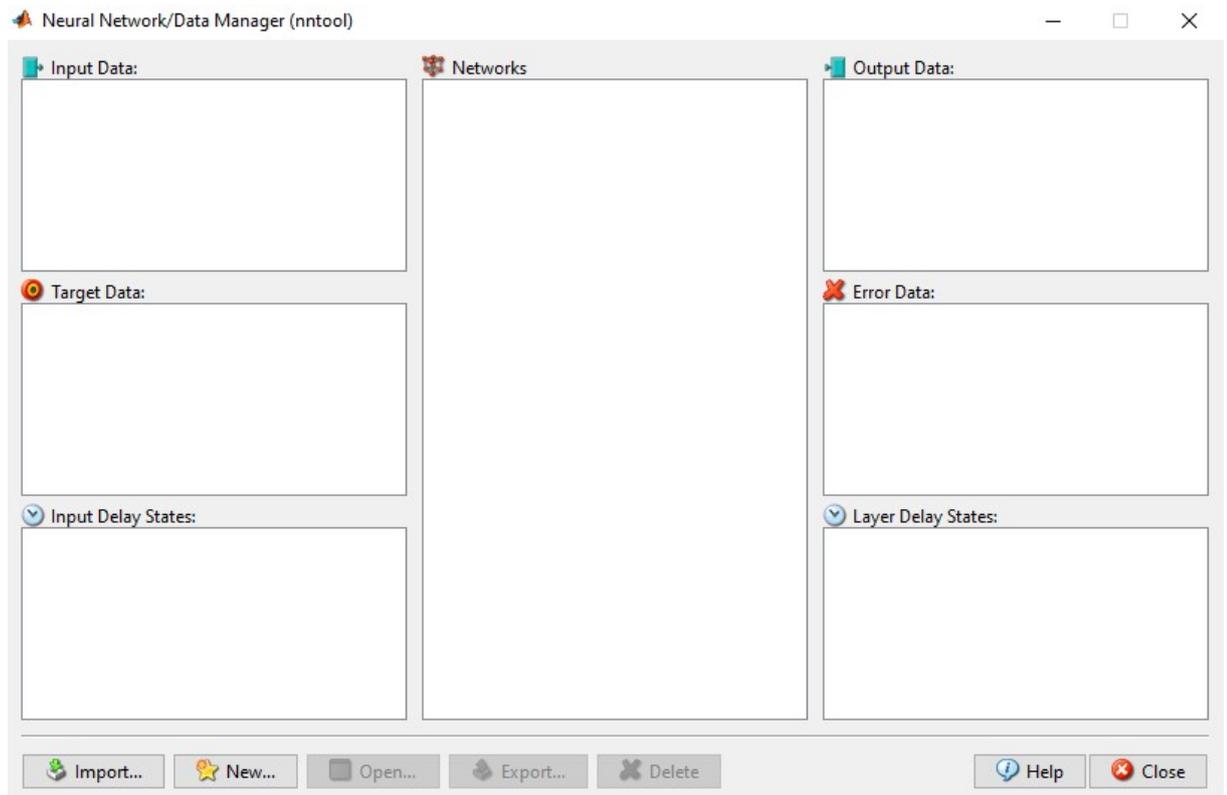


**Figura 7:** Botão de seleção para a pasta da base de dados

Para a preparação das redes MLP usaremos a *toolbox* do *Matlab* (*nntool*) como mostra a **Figura 8**. A seguir é necessário que as entradas e saídas sejam importadas no botão *import* apresentado na **Figura 8**, em seguida a base de dados é selecionada e indicada qual a sua função na rede, se é entrada (*Input Data*) ou saída (*Targets Data*).

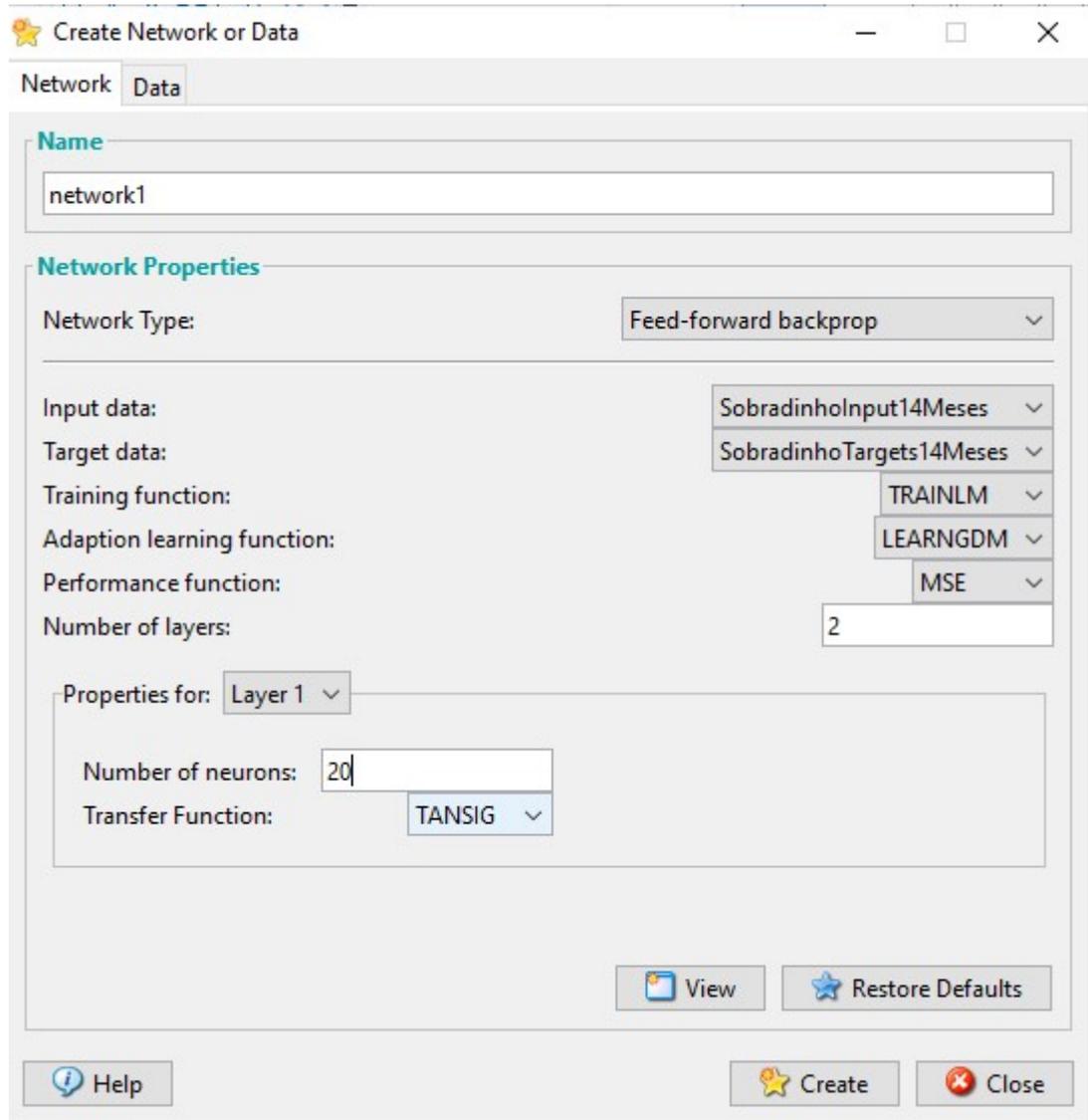
Após essas etapas a *toolbox* irá reconhecer as bases de dados e pode-se criar a nossa rede MLP, para isso clica-se no botão *new* apresentado na **Figura 8**, em seguida uma janela de configuração da rede é apresentada como mostra a **Figura 9**. No campo de *Network type* é selecionado a opção de *Feed-forward backprop* onde indica que usaremos o *Backpropagation* como algoritmo de treinamento para a rede MLP. As bases de dados devem ser selecionadas nos campos de *Input Data* e *Target Data*. Foram utilizadas duas camadas nesse trabalho e selecionou-se isso no campo de *Number of layers*, e no campo de *Number of*

*neurons* indica o numero de neurônios que estará presente na camada escondida, foram utilizadas 14, 20 e 30 neurônios, e o que apresentou melhor coeficiente de correlação foi com 20 neurônios e por isso foi utilizado 20 neurônios na camada escondida. Em seguida a rede é criada através do botão *Create* como apresenta a **Figura 9**.



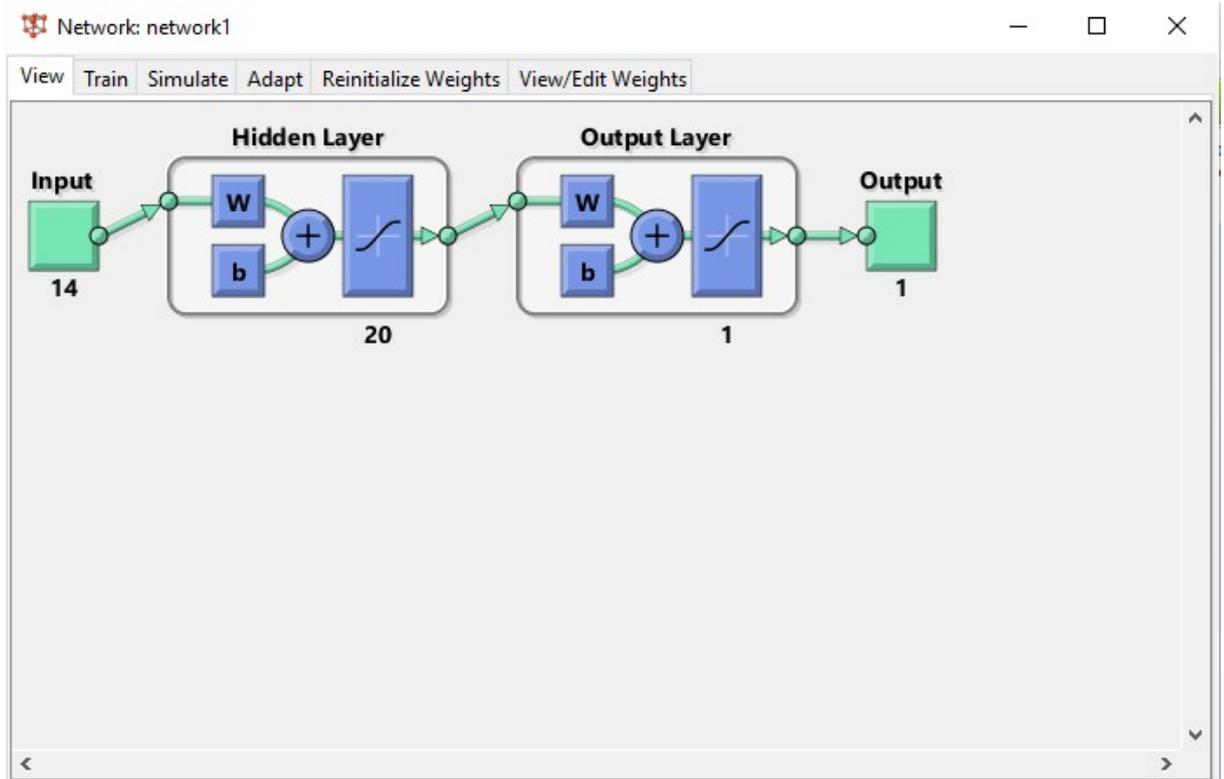
**Figura 8:** *toolbox* do *MatLab*

Ao clicar na rede criada na janela apresentada na **Figura 8**, uma nova janela aberta com as propriedades da rede criada como mostra a **Figura 10**.



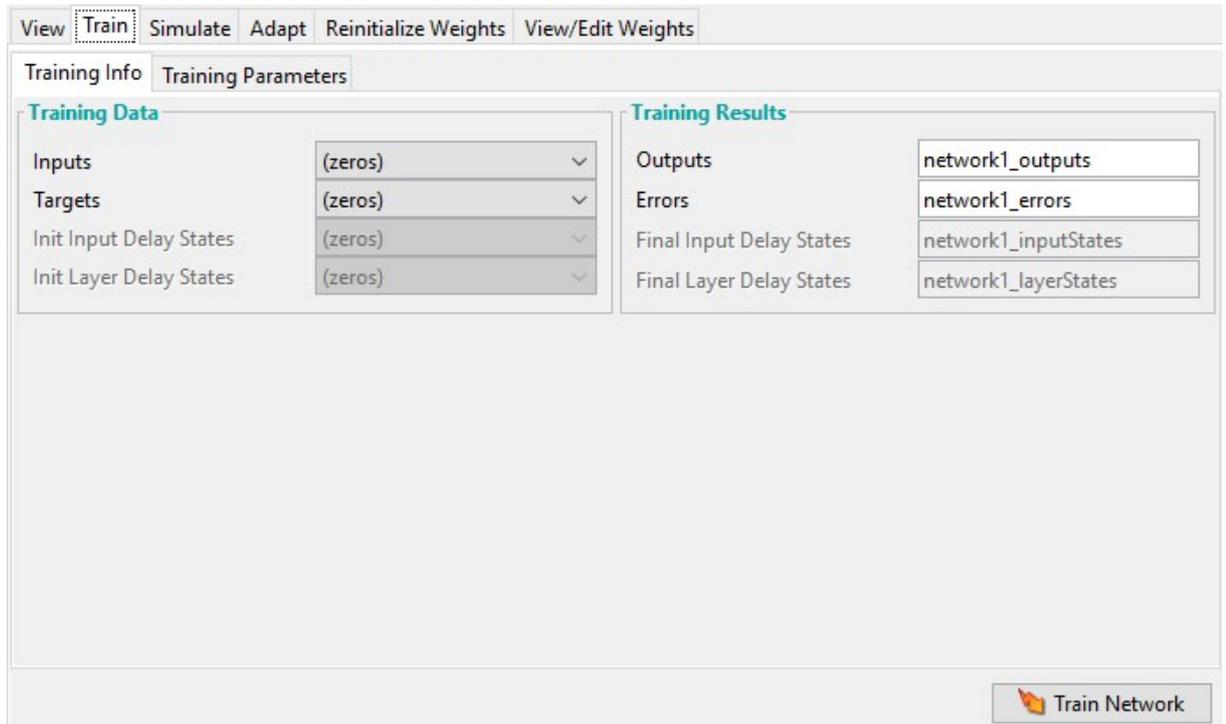
**Figura 9:** Janela de configuração da nova rede MLP

Com a rede criada é necessário que a rede seja treinada, no campo de *Train* da **Figura 10** as entradas e saídas devem ser selecionadas nos campos de *Input* e *Targets* como indica a **Figura 11**. Em seguida a rede deve ser treinada clicando no botão *Train Network* indicada na **Figura 11**.



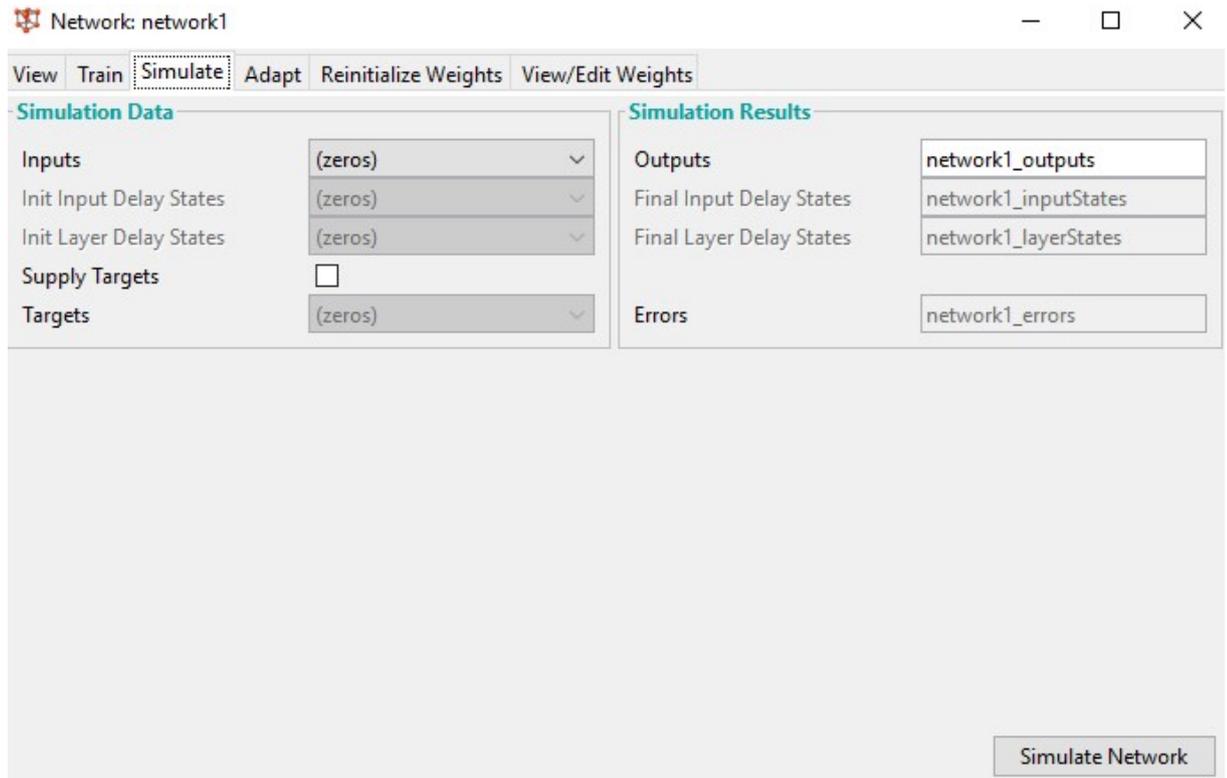
**Figura 9:** Esquema com propriedades da nova rede MLP criada

Após a rede ter sido treinada, serão gerados dois arquivos com os nomes indicados nos campos de *Outputs* e *Errors* como mostra a **Figura 10**, com as saídas e os erros calculados respectivamente, esses arquivos devem ser exportados clicando no botão *Export* apresentado na **Figura 8**.



**Figura 10:** Janela de treinamento da rede MLP

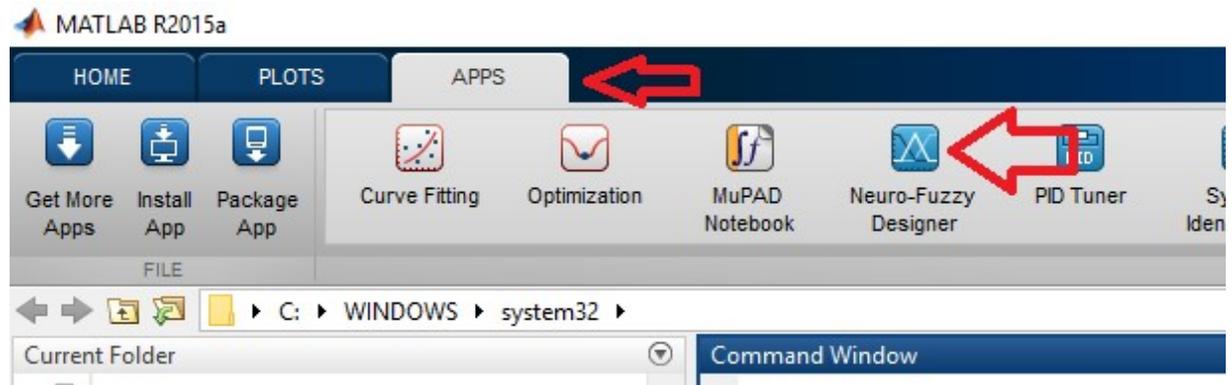
Com isso a rede já pode ser usada clicando no campo de *Simulate* apresentado na **Figura 11**, agora apenas a base de dados de entrada deve ser selecionada no campo de *Inputs* apresentado na **Figura 11** e um novo arquivo com saídas será gerado, como indicado no campo de *Outputs* ainda na **Figura 11**, esse arquivo é exportado. O arquivo gerado tem formato que a *toolbox* produz, a saída como uma linha de arquivo, com isso o comando *transpose* pode ser usado para que se retorne ao formato original da base de dados (saída como coluna do arquivo).



**Figura 11:** Janela de simulação da rede MLP criada e treinada

### 3.2.2 Rede ANFIS

Para criar a uma rede *ANFIS* é necessário acessar a aplicação do *Neuro-Fuzzy Designer* na tela inicial do *Matlab* como mostra a **Figura 12**. Uma janela é carregada permitindo a configuração da rede a ser criada. Importante resaltar que o formato da base de dados usada é diferente da rede MLP. As entradas e saída da rede ANFIS são as colunas do arquivo, sendo a ultima coluna reservada para a saída e todas as demais colunas são as entradas da rede.

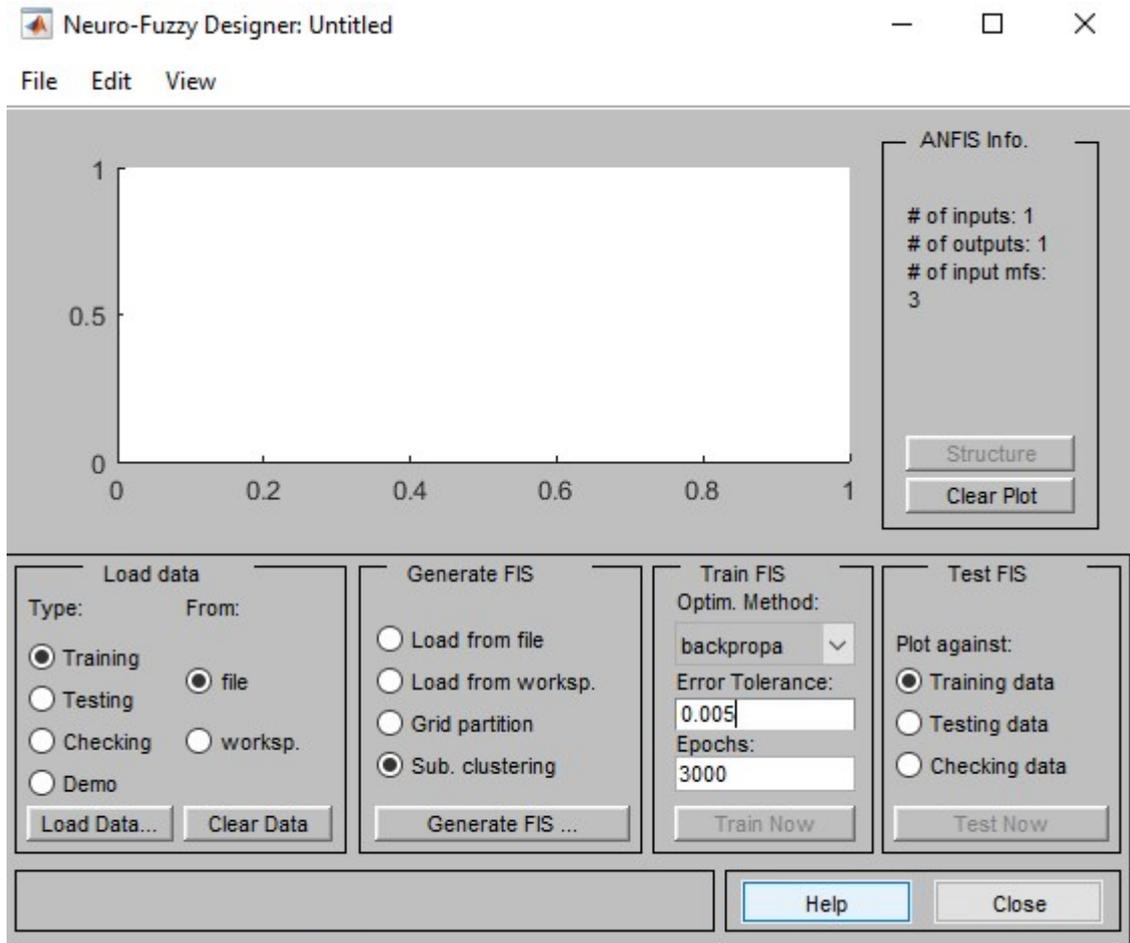


**Figura 12:** Acesso ao painel de criação de uma rede ANFIS

Após o acesso ao *Neuro-Fuzzy Designer* a janela de criação da rede ANFIS será aberta, como mostra a **Figura 13**, logo abaixo existem 4 painéis que devem ser configurados, primeiro painel *Load data*, deve ser carregado a base de dados que usaremos para treinar a rede ANFIS, para isso a opção de *Training* deve ser marcada e o arquivo deve ser carregado; o segundo o painel *Generate FIS* é a forma que as regras da rede são geradas, a opção *Sub. clustering* que usa gaussianas para gerar as regras da rede;

O terceiro painel o *Train FIS* é a configuração do treinamento da rede, utiliza-se o *back propagation* e os dois parâmetros seguintes, *Error tolerance* onde estima-se um erro aceitável para a rede, sendo que se a rede atingir o erro indicado o treinamento encerrado, para esse trabalho usa-se um erro de tolerância muito baixo onde, pelas características do domínio do problema, não esperava-se que fosse atingido, e foi exatamente o que aconteceu, essa medida foi tomada para que fosse visualizado o grau de aprendizagem da rede de acordo com o passar das épocas de treinamento. O último parâmetro do mesmo campo indica o número de épocas que a rede será treinada no máximo, usamos um valor muito grande de épocas (para esse trabalho 3000) para que se visualizasse até onde valeria a pena continuar treinando a rede com clareza.

O quarto e último campo o *Test FIS* é o campo de uso propriamente dito da rede, onde é possível usar a própria base de dados de treinamento para uso e teste da rede.



**Figura 13:** Painel de criação e uso da rede ANFIS

A arquitetura da rede acessada e as regras geradas podem ser acessadas pela opção *Edit* como indica a **Figura 14**, a opção *Rules...* acessa as regras, ou seja, o conjunto de gaussianas que estarão operando diretamente as entradas, e a opção *Anfis...* que acessa a arquitetura gerada, a arquitetura não muda, serão as mesmas 5 camadas como visto anteriormente, a diferença é que o número de regras é visto com maior clareza.

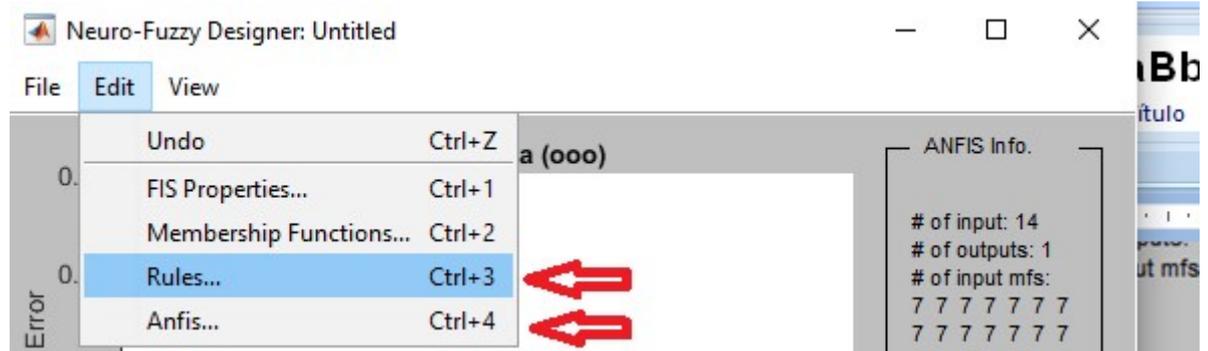


Figura 14: Acesso às regras e à arquitetura

### 3.3 Métricas de performance

A primeira métrica de performance que foi adotado para esse trabalho é o coeficiente de correlação e utilizado na regressão é o  $r^2$ . Também chamado de de determinação:

$$r^2 = \frac{(n \sum_{i=1}^n f(x_i)y_i - \sum_{i=1}^n f(x_i) \sum_{i=1}^n y_i)^2}{(n \sum_{i=1}^n f(x_i)^2 - (\sum_{i=1}^n f(x_i))^2)(n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2)}$$

que indica quanto da variância da variável resposta  $f(x_i)$ , é explicada pela variância das variáveis explicativas  $y_i$ , um valor que mede o grau de relacionamentos entre as duas variáveis. Seu valor está no intervalo [0,1].

Utiliza-se também o erro quadrático médio o EQM, expressado pela equação:  $EQM(\hat{\theta}) = E(\hat{\theta} - \theta)^2$ ; onde o operador  $E$  denota a operação do valor esperado ou esperança [8] [9]. Os parametros  $\theta$  e  $\hat{\theta}$  são o valor esperado e o valor que o sistema enviou respectivamente.

E também o desvio padrão, ele mostra a dispersão ou a quantidade de variação dos resultados obtidos em relação à média. Pode ser representado pela seguinte equação:  $\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$ , onde  $x_i$  é a saída e o  $\bar{x}$  é a média.

# Capítulo 4

## Resultados

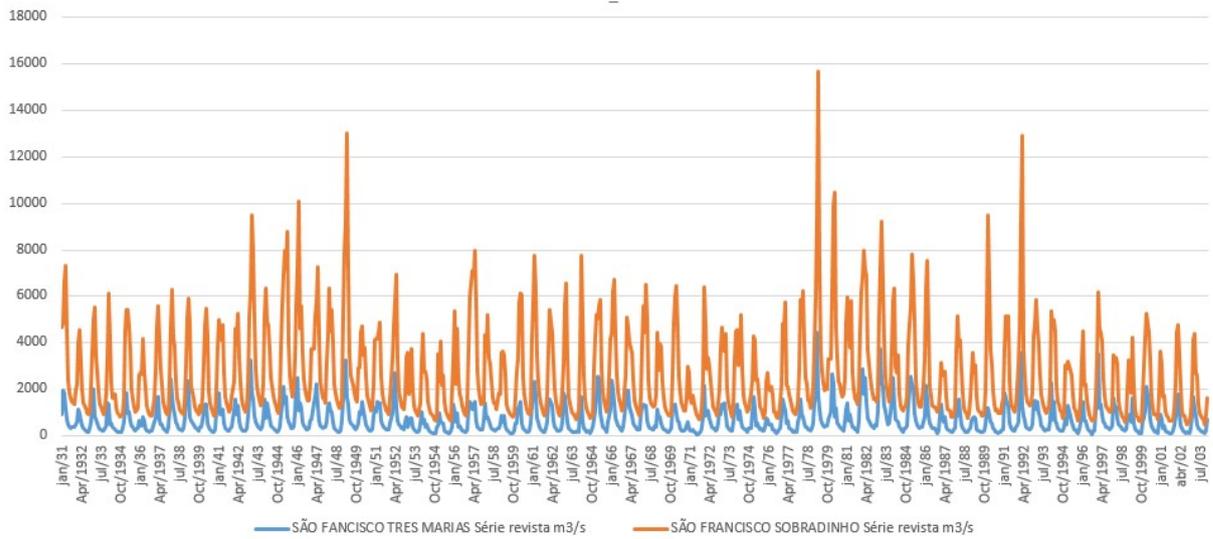
Neste capítulo é apresentado como os experimentos foram realizados no treinamento das redes e os resultados obtidos. Na seção 4.1 é descrita a metodologia utilizada na previsão, e na seção 4.2 são apresentados os experimentos realizados, os resultados e comparações dos resultados.

### 4.1 Metodologia

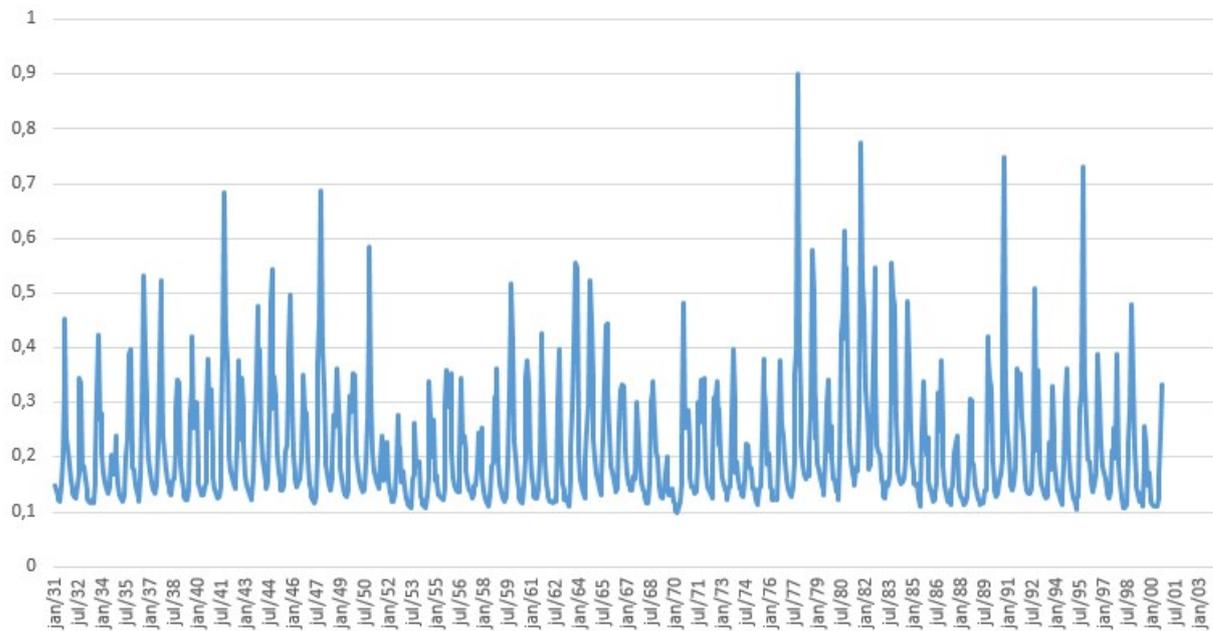
Este trabalho tem como objetivo apresentar um método de previsão de vazões mensais tendo como entrada vazões de quatorze meses anteriores, com horizonte máximo de um mês, a ser utilizado na avaliação de desempenho.

A estratégia é comparar os resultados obtidos, os valores de erro quadrático médio para um mês à frente.

A primeira etapa do processo está relacionada ao pré-processamento das bases de dados, onde houve um tratamento das séries. Aplicou-se a regra de normalização descrita na seção 3 e mostrada graficamente na **Figura 15**, antes da normalização, e depois, na **Figura 16** e **Figura 17**.



**Figura 15:** Série original das Usinas de Sobradinho e Três Marias, sem normalização dos dados.



**Figura 16:** Série da Usina Três Marias, após a normalização.

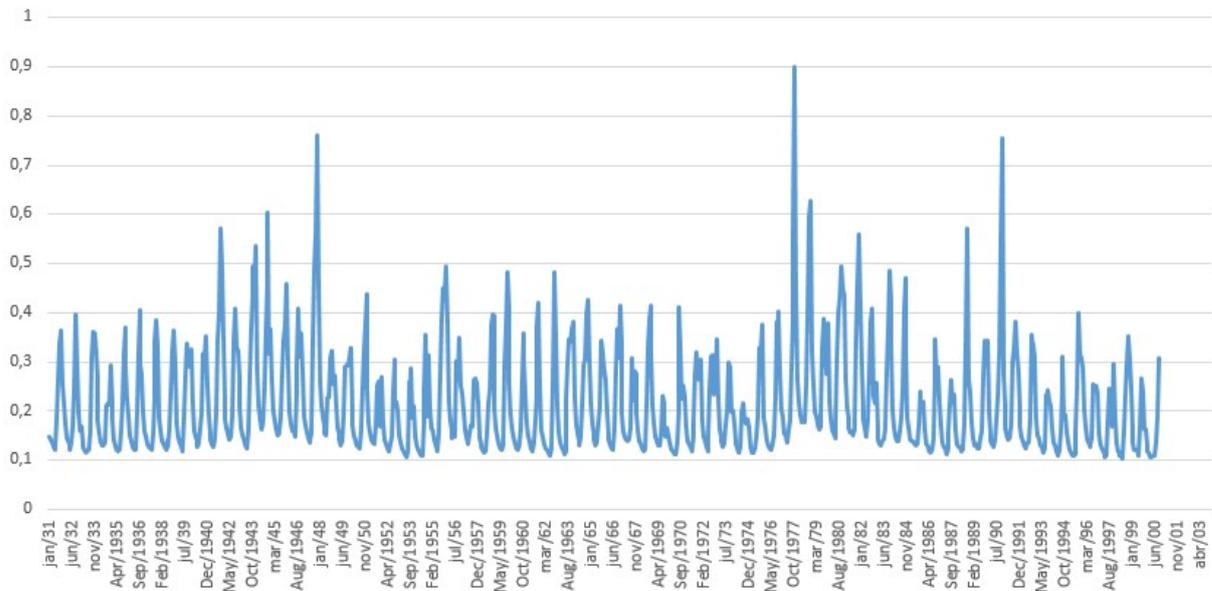


Figura 17: Série da Usina Sobradinho, após a normalização.

## 4.2 Experimentos e Resultados

### 4.2.1 Experimentos realizado com a rede MLP

A segunda etapa foi modelar a rede MLP, para isso teve-se 4 valores para o número de neurônios da camada escondida, sendo escolhido o valor com o maior R, sendo testadas com 14, 20 e 30 neurônios na camada escondida, com 14 neurônios obtivemos o maior R no geral para as etapas de treinamento como apresenta a na **Tabela 1** e na **Figura 18** um esquema simplificado de blocos da arquitetura da rede.

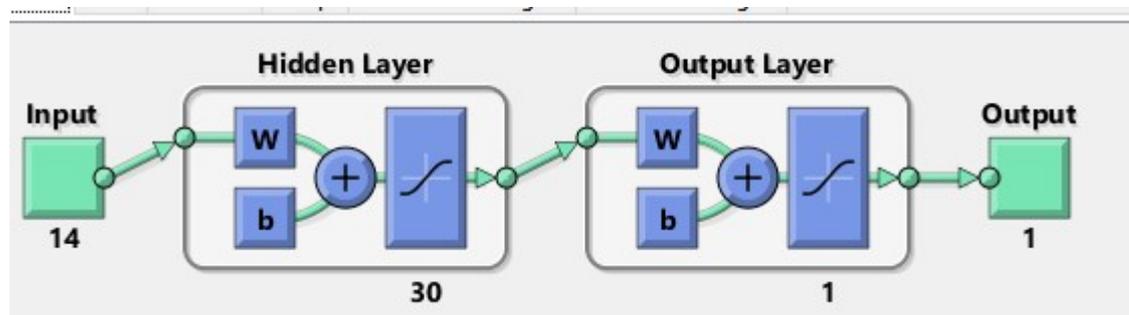
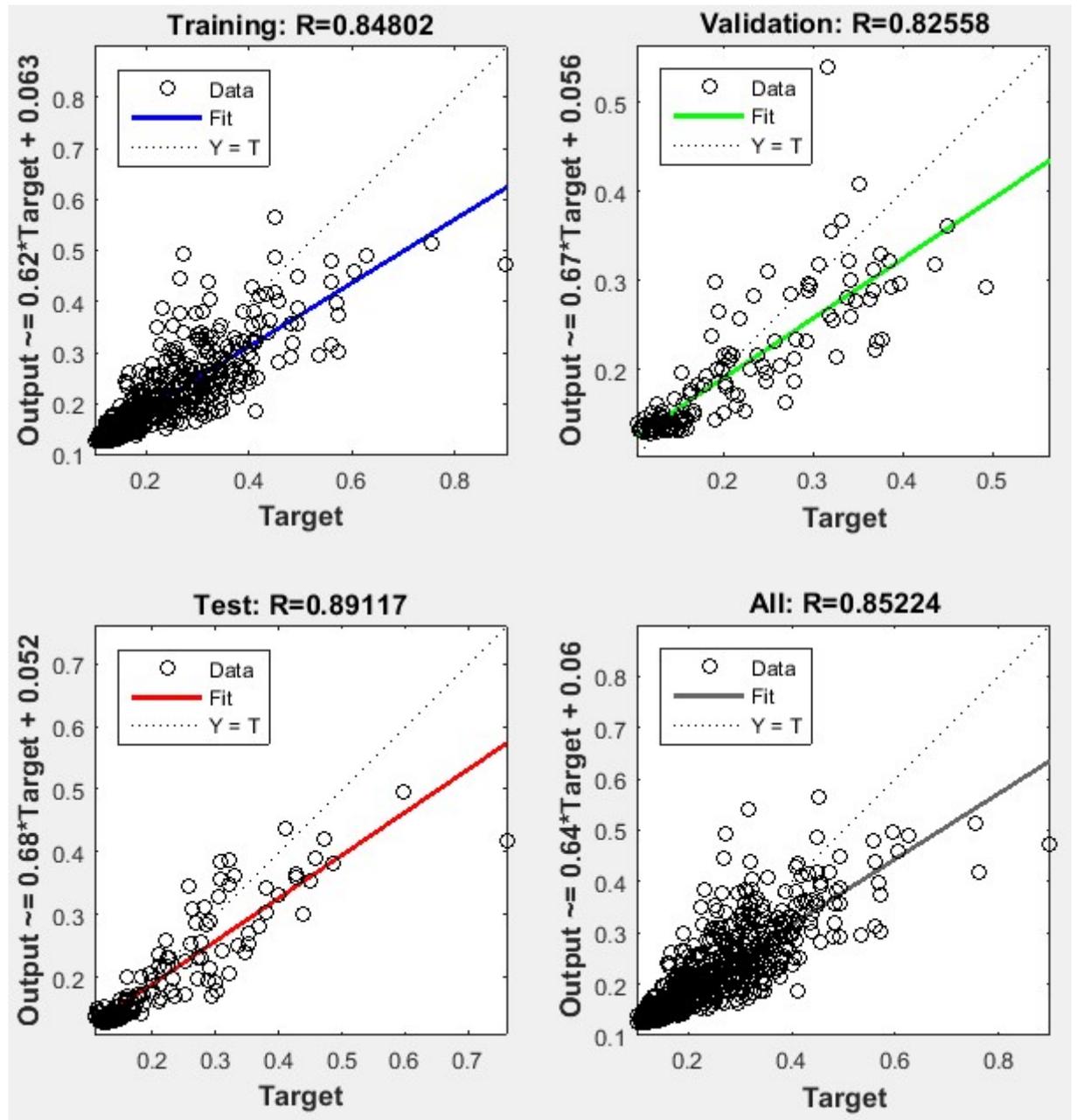
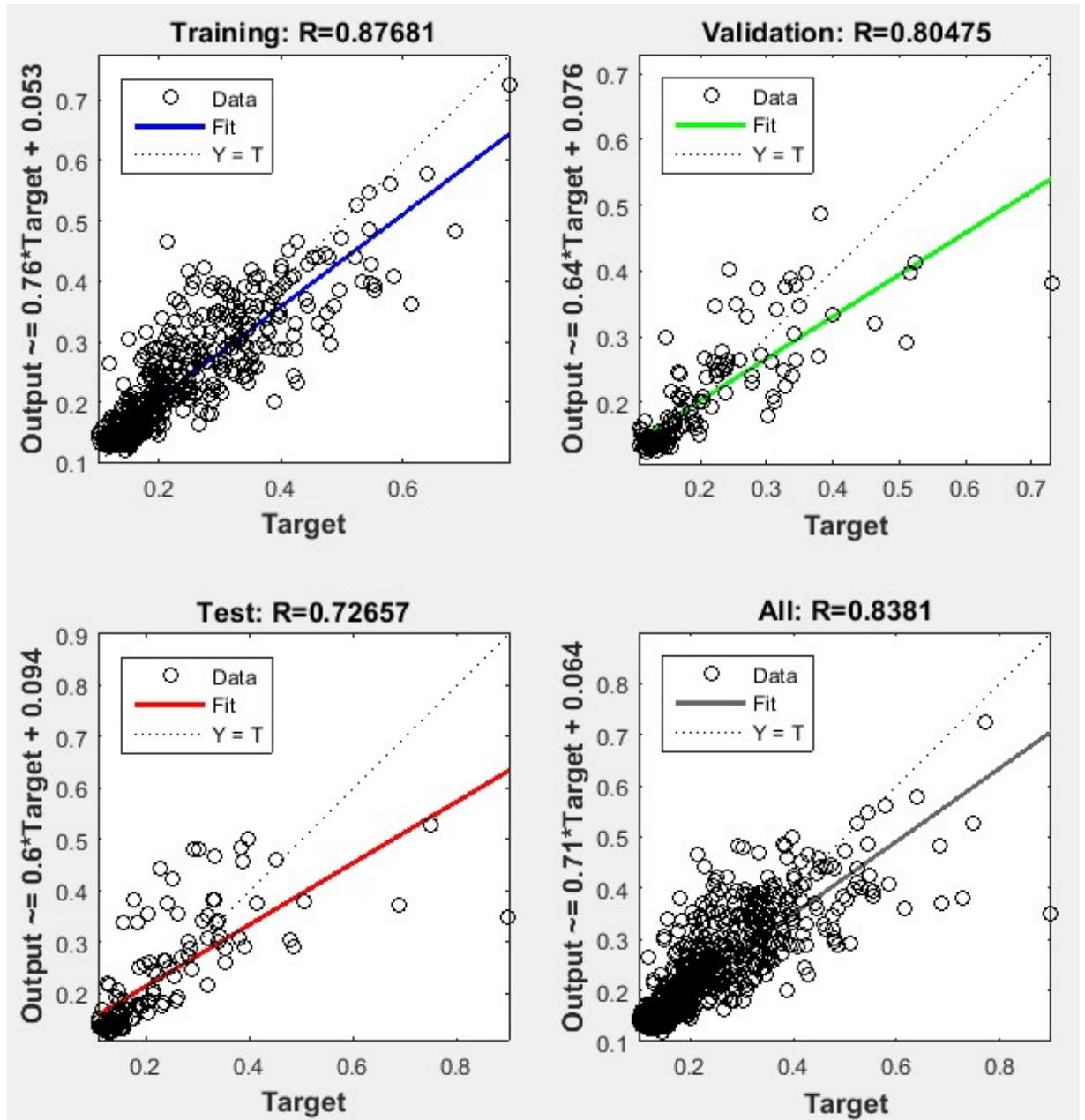


Figura 18: Esquema de blocos simplificado da rede MLP

A rede foi treinada usando o algoritmo *back-propagation* e obtém-se o R juntamente com as suas curvas de calibração como indica a **Figura 19** para a vazão de Sobradinho e na **Figura 20** para Três Marias.

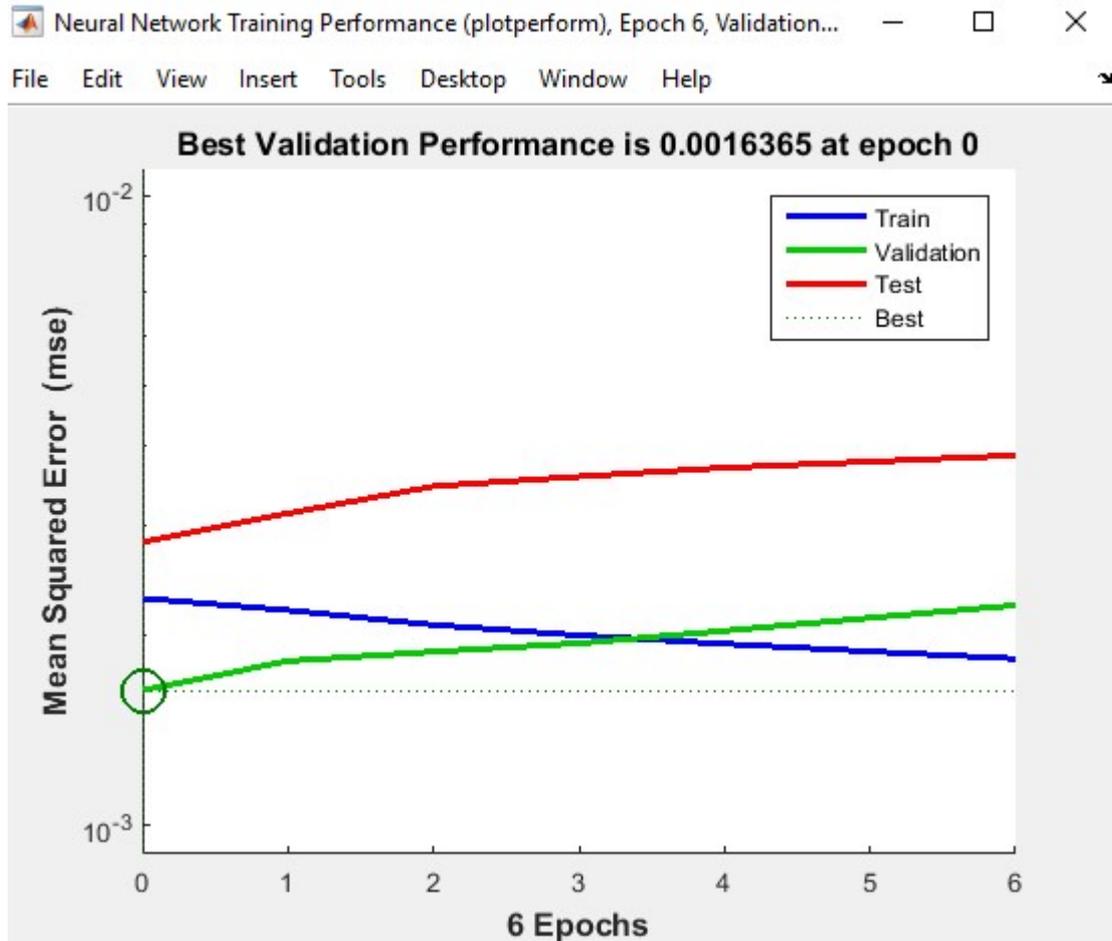


**Figura 19:** Regressão para a usina de Sobradinho

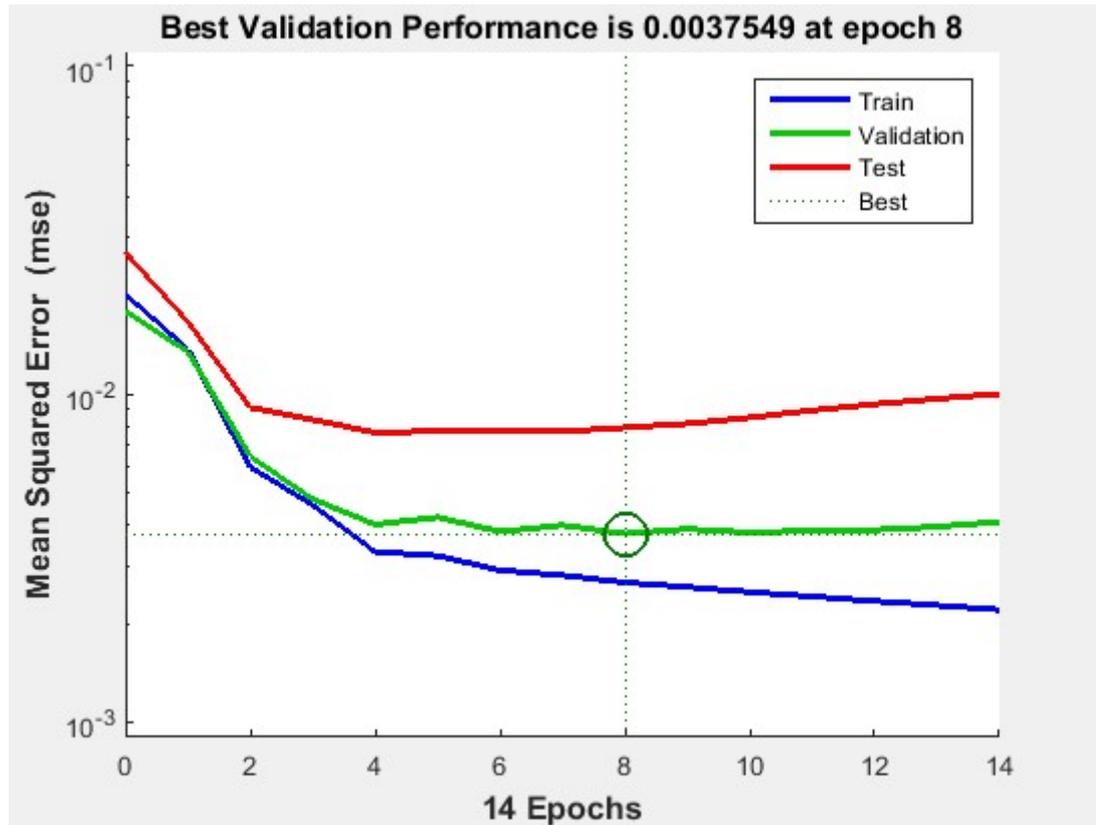


**Figura 20:** Regressão para a usina de Três Marias

Observa-se que a melhor época de treinamento para as redes são de 0 (nenhuma) época para Sobradinho como indica a **Figura 21** e de 14 épocas para a usina de Três Marias como apresentado na **Figura 22**.



**Figura 21:** Melhor validação de performace para Sobradinho



**Figura 22:** Melhor validação de performace para Três Marias

**Tabela 1:** Resultados dos R para Sobradinho

	Treino	Validação	Testes	Todos
14 neuronios	0,84802	0,82558	0,89117	0,85224
20 neuronios	0,85878	0,81764	0,82087	0,84407
30 neuronios	0,87301	0,7529	0,68916	0,82748

**Tabela 2:** Resultados dos R para Três Marias

	Treino	Validação	Testes	Todos
14 neuronios	0,82793	0,77615	0,80129	<b>0,81661</b>
20 neuronios	0,84223	0,76213	0,71347	0,80162
30 neuronios	0,81342	0,75231	0,80614	0,81041

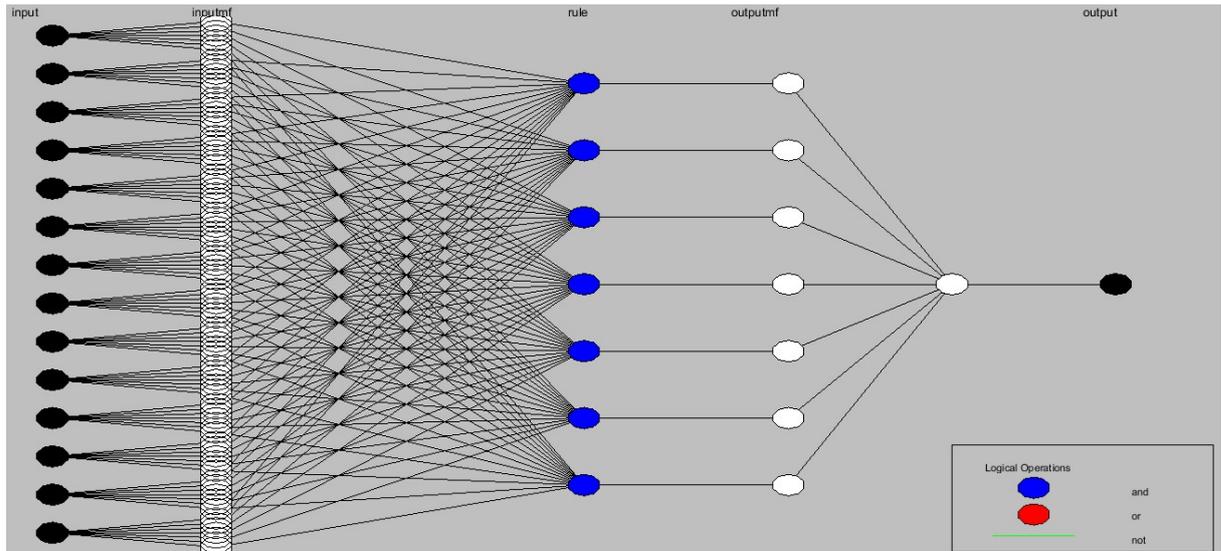
Após os experimentos verificou-se que a rede com um menor erro quadrático médio, para isso usamos a rede com 14 neurônios na camada escondida e foi variada a quantidade de entradas (meses) em: 14, 12, 6 e 2 meses, a rede que possuísse o melhor resultado seria selecionada para a comparação com a rede *ANFIS*. Para ambos os casos a rede com 14 meses como entrada mostrou um erro quadrático médio melhor do que as outras opções, como indica a **Tabela 3**.

**Tabela 3:** Erro Quadrático Medio para ambas Sobradinho e Três Marias

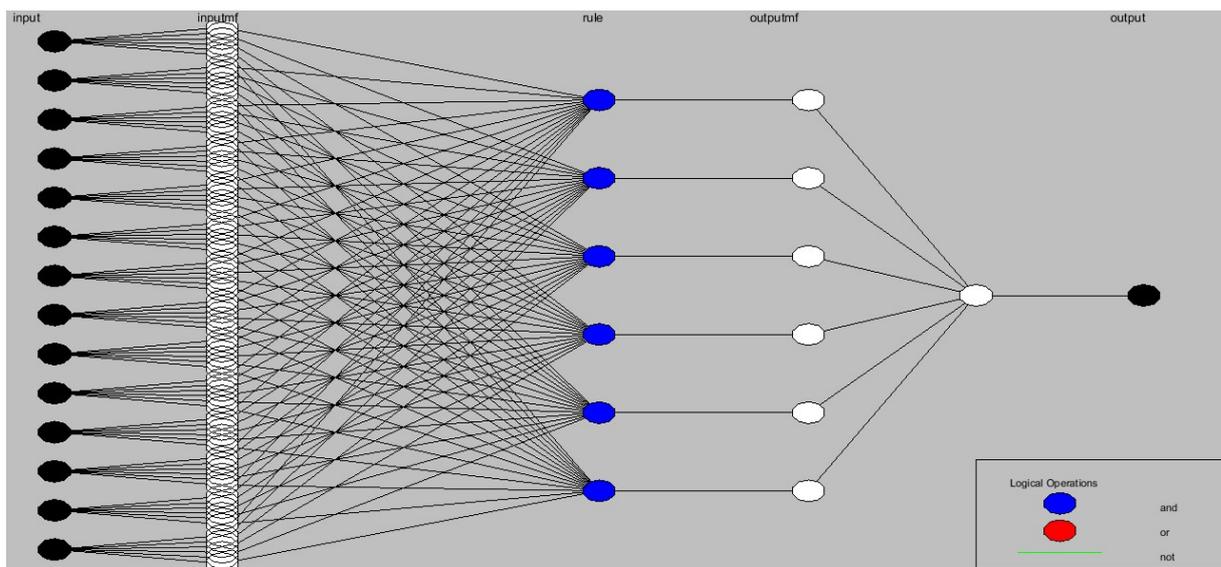
	14 meses	12 meses	6 meses	2 meses
Sobradinho	<b>0,002548642</b>	0,002890816	0,003206248	0,002958528
Três Marias	<b>0,004175953</b>	0,004358283	0,004302005	0,004815555

#### 4.2.2. Experimentos realizado na rede ANFIS

Para a rede neuro fuzzy escolhida o processo foi muito semelhante, foi escolhida 14 meses de antecedência para prever um mês e assim mantendo a igualdade nas condições das redes e dessa maneira fazer possíveis comparações, para a Sobradinho a arquitetura gerada apresentou 7 regras(gaussianas) na arquitetura para cada uma das entradas como mostra a **Figura 23**, e de 6 regras na arquitetura para Três Marias como apresentado na **Figura 24**.



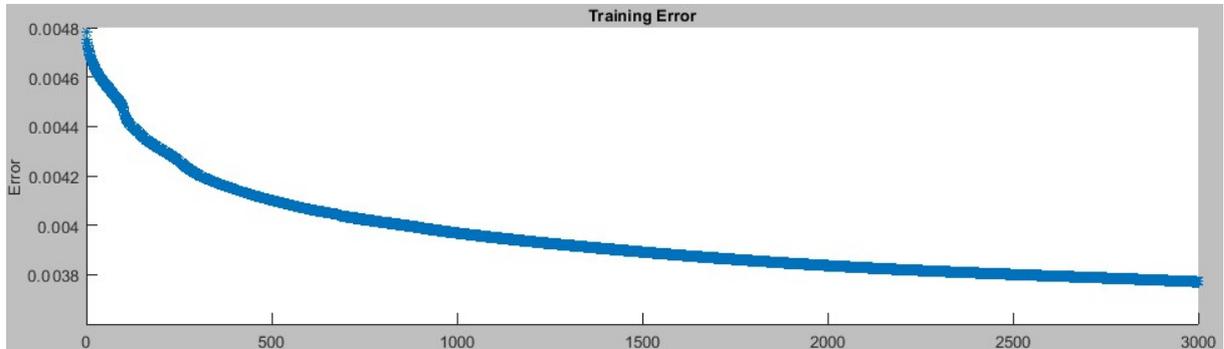
**Figura 23:** Arquitetura ANFIS para Sobradinho



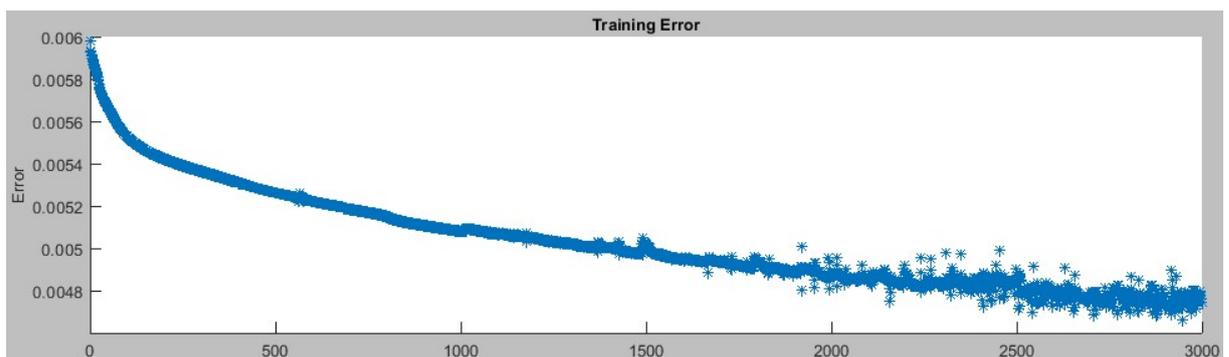
**Figura 24:** Arquitetura ANFIS para Três Marias

O processo de treinamento foi executado, sendo alimentado pelos seguintes parâmetros: 0.0005 de erro de tolerância e 3000 épocas de limite de treinamento. A justificativa para os parâmetros é a seguinte, foi optado por um erro muito baixo, ao qual não se espera que a rede atinja e assim se veja a curva de erro mediante a um número excessivo de épocas para que se veja claramente onde não se possui

ganhos significativos com o treinamento. A curva de erro para os dados de Sobradinho e Três marias estão presentes na **Figura 25** e na **Figura 26** respectivamente.

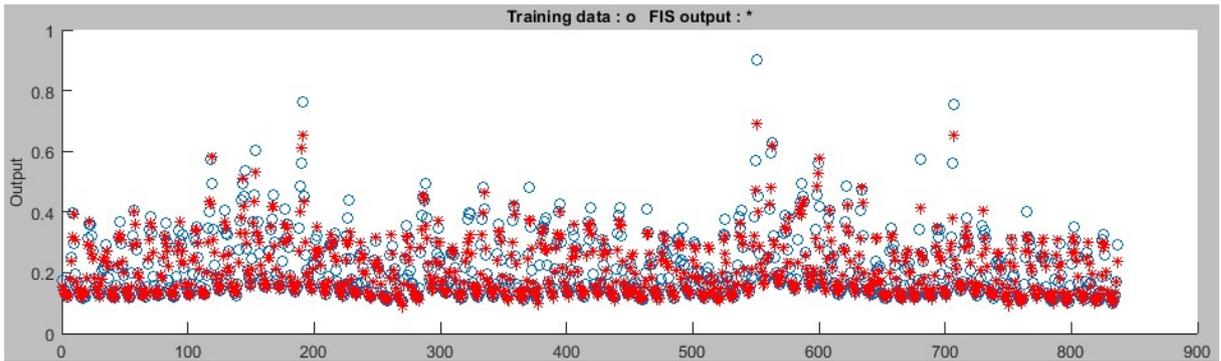


**Figura 25:** Curva de erro para os dados de Sobradinho

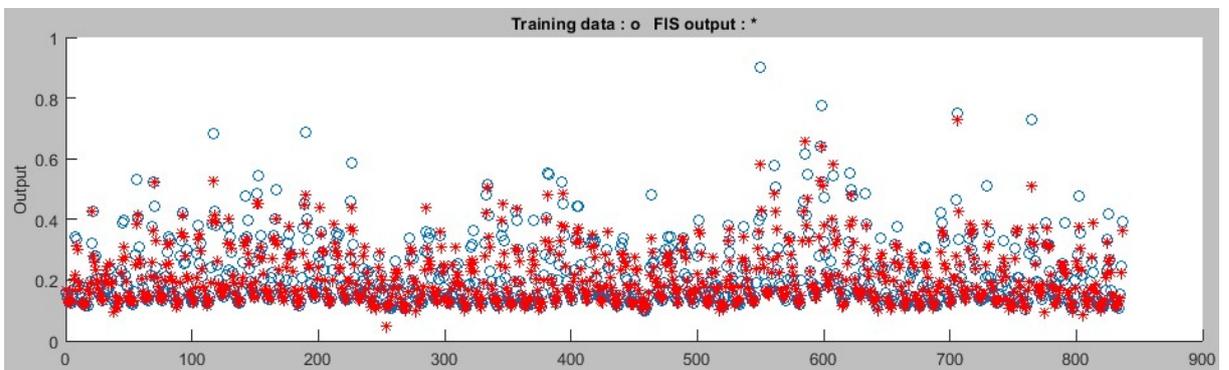


**Figura 26:** Curva de erro para os dados de Três Marias

Após o treinamento as redes foram executadas e em seguida na **Figura 27** e na **Figura 28** são apresentados graficamente os resultados da execução da rede para Sobradinho e Três Marias respectivamente, onde os pontos vermelhos são a saída e os círculos azuis são o esperado, e na **Tabela 4** os erros e na **Tabela 5** o desvio padrão para cada um dos casos.



**Figura 27:** Resultado da execução da rede ANFIS para Sobradinho



**Figura 28:** Resultado da execução da rede ANFIS para Três Marias

**Tabela 4:** Comparação do EQM entre as redes MLP e ANFIS

	MLP	ANFIS
Sobradinho	0,002958528	0,002365786
Três Marias	0,004815555	0,003998547

**Tabela 5:** Desvio padrão para as redes MLP e ANFIS

	MLP	ANFIS
Sobradinho	0,093929119	0,087928468
Três Marias	0,081199185	0,073294650

# Capítulo 5

## Conclusão

Neste trabalho, foi abordado o problema de previsão de vazões para hidrelétricas. Objetivando uma melhoria de resultado dos sistemas atuais utilizados, foi proposto o uso de uma rede neural que utilizasse o paradigma da lógica *fuzzy*. A rede escolhida foi a rede *ANFIS*.

Os resultados da comparação dos resultados das hidrelétricas foi objetivo, sendo que qual quer melhora nos resultados já existentes é um bom avanço para problemas dessa natureza. Para as redes comparadas a rede *ANFIS* obteve um erro quadrático médio e uma concentração de resultados um pouco melhor que a rede tradicional porém os resultados são muito semelhantes sendo necessário mais testes estatísticos para a comprovação de seu melhor desempenho. Porém o domínio do problema de previsões é muito variável, sendo que é possível garantir um melhor resultado para os casos abordados para este trabalho, não sendo necessariamente aplicados a outros casos, como mostrado as bases de dados têm características muito próprias que as torna muito diferentes, para outras bases de dados é possível que a rede *ANFIS* não possua um desempenho melhor do que a rede clássica utilizadas.

Para trabalhos futuros é possível fazer comparações entre um maior número de modelos testando-se um maior número de parâmetros estatísticos, e até propor uma nova arquitetura para o paradigma neuro-fuzzy.

# Bibliografia

- [1] Norvig, Peter; Russel, Stuart. *Inteligência Artificial* 2 ed. Elsevier [S.l.], 2004
  - [2] ALMEIDA, P. E. M.; EVSUKOFF, A. G. Sistemas fuzzy. In: REZENDE, S. O. (Ed.). *Sistemas Inteligentes: Fundamentos e Aplicações*. [S.l.]: Manole, 2005.
  - [3] SANDRI, S.; CORREA, C. Lógica nebulosa. *Anais da V Escola de Redes Neurais*.
  - [4] Ahlawat, Nishant, Ashu Gautam, and Nidhi Sharma (International Research Publications House 2014) "Use of Logic Gates to Make Edge Avoider Robot." *International Journal of Information & Computation Technology* (Volume 4, Issue 6; page 630) ISSN 0974-2239 (Retrieved 27 April 2014)
  - [5] KASABOV, N.K. *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*. A Bradford Book, London, 1996.
  - [6] JANG, J.-S. R. Anfis: Adaptive-network-based fuzzy inference systems. *IEEE Trans. on Systems, Man, and Cybernetics*
  - [7] Tan, P., Steinbach, M., & Kumar, V.. *Introdução ao Data Mining*. Rio de Janeiro: Ciência Moderna, 2009.
  - [8] Propriedades dos Estimadores. icmc. Consultado em 23 de Outubro de 2016.
  - [9] Introdução aos Processos Estocásticos - Estimadores. PPGEE - Universidade Federal de Minas Gerais. Consultado em 23 de Outubro de 2016.
  - [10] World Wildlife Fund. Washington, DC. Acessado em 6 de Novembro de 2016. Disponível em: <http://www.worldwildlife.org/threats/water-scarcity>.
-

- [11] DAVIES, R. World Disasters Report – Most Deaths Caused by Floods. Acessado em 6 de Novembro de 2016. Disponível em: <http://floodlist.com/dealing-with-floods/world-disastersreport-100-million-affected-2013>.
- [12] Agência Nacional de Energia Elétrica. Brasília, Brasil. BIG – Banco de Informações de Geração – Capacidade de Geração do Brasil. Acessado em 6 de Novembro de 2016. Disponível em: <http://www.aneel.gov.br/aplicacoes/capacidadebrasil/capacidadebrasil.cfm>.
- [13] ONS. (Outubro de 2011). Acesso em 28 de Setembro de 2011, disponível em PROGRAMA MENSAL DA OPERAÇÃO (PMO) RELATÓRIO MENSAL DE PREVISÃO DE VAZÕES E GERAÇÃO DE CENÁRIOS DE AFLUÊNCIAS: <http://www.ons.gov.br/>
-