

UNIVERSIDADE DE PASSO FUNDO

PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA CIVIL E AMBIENTAL

Área de concentração: Infraestrutura e Meio Ambiente

Dissertação de Mestrado

MODELO DE REDES NEURAIAS ARTIFICIAIS PARA  
PREVISÃO DE FREQUÊNCIA NATURAL EM LAJES DE  
CONCRETO ARMADO

Bruno Marangoni

Passo Fundo

2021



CIP – Catalogação na Publicação

---

M311m Marangoni, Bruno  
Modelo de redes neurais artificiais para previsão de  
frequência natural em lajes de concreto armado / Bruno  
Marangoni. – 2022.  
61 f. : il. color. ; 30 cm.

Orientador: Prof. Dr. Moacir Kripka.  
Dissertação (Mestre em Engenharia) – Universidade de  
Passo Fundo, 2022.

1. Redes neurais (Computação). 2. Dinâmica estrutural.  
3. Lajes de concreto. I. Kripka, Moacir, orientador. II. Título.

CDU: 624.012.35

---

Catálogo: Bibliotecário Luís Diego Dias de S. da Silva – CRB 10/2241

Bruno Marangoni

MODELO DE REDES NEURAIAS ARTIFICIAIS PARA PREVISÃO  
DE FREQUÊNCIA NATURAL EM LAJES DE CONCRETO  
ARMADO

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil e Ambiental, da Faculdade de Engenharia e Arquitetura da Universidade de Passo Fundo, como requisito para obtenção do título de Mestre em Engenharia, sob a orientação do Prof. Dr. Moacir Kripka.

Data da aprovação: 28/04/2022

Membros componentes da Banca Examinadora:

Doutor Moacir Kripka

Orientador

Doutor Zacarias Martin Chamberlain Pravia

Universidade de Passo Fundo – UPF

Doutor Francisco Dalla Rosa

Universidade de Passo Fundo – UPF

Doutor Herbert Martins Gomes

Universidade Federal do Rio Grande do Sul – UFGRS

Passo Fundo

2022

## RESUMO

Nos projetos de estruturas é necessário analisar as ações externas dinâmicas e determinar o efeito e a relevância das vibrações causadas, para que se possa ter controle, por exemplo, das fissurações excessivas e da forte vibração da estrutura, com base no dimensionamento adequado dos componentes estruturais ou pelo limite determinado pela frequência natural. As metodologias de obtenção de frequência natural demandam conhecimento técnico específico, são de elevado custo financeiro em virtude dos equipamentos e softwares, e exigem tempo considerável para execução. Com o objetivo desenvolver uma ferramenta alternativa para mensurar essa propriedade nas estruturas de concreto de forma simplificada, reduzindo o tempo de execução e os custos com os ensaios, foi proposto estudar e elaborar um modelo de previsão de frequências naturais das lajes utilizando Redes Neurais Artificiais (RNA). Para isso, foram elaboradas arquiteturas de rede de acordo com o banco de dados compostos por leituras *in loco* de lajes nervuradas e pré-moldadas. O treinamento supervisionado, validação e teste das redes foi realizado com as arquiteturas (11-6-1) e (8-5-1), respectivamente. Dentre os hiperparâmetros analisados, o *earlystopping* foi responsável pela parada do treinamento no ponto onde a rede apresentou o melhor desempenho, conseqüentemente o menor valor de erro por meio das métricas utilizadas. As redes puderam ser testadas com 30% dos dados de entrada, comparando-os com as previsões realizadas por elas. É possível afirmar que a rede possibilitou a previsão de frequências de forma satisfatória para as lajes de baixas frequências. Para as lajes com frequências mais elevadas a rede não foi capaz de prever de forma precisa, uma vez que se detectou uma necessidade de manipulação de dados faltantes no banco de dados.

Palavras-chave: Redes Neurais Artificiais; Vibrações; Frequência Natural; Estruturas; Laje de Concreto Armado.

## ABSTRACT

In structural designs, it is necessary to analyze the dynamic external actions and determine the effect and relevance of the vibrations caused, so that, for example, excessive cracking and strong vibration can be controlled, based on the proper sizing of the structural components or by the limit determined by the natural frequency. The methodologies to obtaining natural frequency demand specific technical knowledge, are of high financial cost due to the equipment and software and require considerable time for execution. With the objective of developing an alternative tool to measure this property in concrete structures in a simplified way, reducing the execution time and the costs with the tests, it is proposed to study and elaborate a model to predict the natural frequencies of the slabs using Artificial Neural Networks (ANN). For this, network architectures were elaborated according to the database composed by *in loco* readings of ribbed and precast slabs. The supervised training, validation and testing of the networks were carried out with the (11-6-1) and (8-5-1) architectures, respectively. Among the hyperparameters analyzed, early stopping was responsible for stopping the training at the point where the network presented the best performance, consequently the lowest error value through the metrics used. The networks could be tested with 30% of the input data, comparing them with their predictions. It is possible to state that the network made it possible to predict frequencies satisfactorily for the low-frequency slabs. For slabs with higher frequencies, the network was not able to accurately predict, since a need to manipulate missing data in the database was detected.

**Keywords:** Artificial Neural Networks; Vibrations; Natural Frequency; Structures; Reinforced Concrete Slab.

## LISTA DE FIGURAS

Figura 1 - Sistema de laje nervurada bidirecional: (a) representação em planta e respectivo (b) corte transversal .....	16
Figura 2 - Sistema de laje pré-moldada treliçada: (a) indicação dos principais elementos que compõem a laje e (b) corte transversal .....	17
Figura 3 - Neurônio artificial genérico .....	21
Figura 4 - Funcionamento da RNA na fase de treinamento .....	23
Figura 5 - Fluxograma de desenvolvimento da RNA .....	28
Figura 6 - Arquitetura de lajes nervuradas proposta da RNA .....	42
Figura 7 - Arquitetura de lajes pré-moldadas proposta da RNA .....	43
Figura 8 - Dados csv .....	44
Figura 9 - Dados normalizados para lajes nervuradas pelo <i>minmaxscaler</i> .....	44
Figura 10 - Dados normalizados para lajes pré-moldadas pelo <i>minmaxscaler</i> .....	45
Figura 11 – Lajes nervuradas: Evolução do MSE em 300 épocas .....	46
Figura 12 - Lajes nervuradas: Evolução do MAE em 300 épocas .....	46
Figura 13 - Lajes nervuradas: Predição da rede na fase de teste com 300 épocas .....	47
Figura 14 - Lajes nervuradas: Evolução do MAE em 1000 épocas .....	48
Figura 15 - Lajes nervuradas: Evolução do MSE em 1000 épocas .....	48
Figura 16 - Lajes nervuradas: Predição da rede na fase de teste com 1000 épocas .....	49
Figura 17 - Lajes nervuradas: Evolução do MAE em 3000 épocas .....	49
Figura 18 - Lajes nervuradas: Evolução do MSE em 3000 épocas .....	50
Figura 19 - Lajes nervuradas: Predição da rede na fase de teste com 3000 épocas .....	50
Figura 20 - Lajes pré-moldadas: Evolução do MAE em 3000 épocas .....	51
Figura 21 - Lajes pré-moldadas: Evolução do MSE em 3000 épocas.....	51
Figura 22 - Lajes pré-moldadas: Predição da rede na fase de teste com 3000 épocas .....	52

## SUMÁRIO

1	INTRODUÇÃO .....	10
1.1	OBJETIVOS .....	12
1.1.1	Objetivo Geral .....	12
1.1.2	Objetivos Específicos .....	12
2	REVISÃO BIBLIOGRÁFICA .....	13
2.1	Sustentabilidade das estruturas de concreto .....	13
2.2	Lajes de concreto armado .....	15
2.2.1	Lajes nervuradas .....	15
2.2.2	Lajes pré-moldadas .....	16
2.3	Dinâmica das estruturas .....	17
2.3.1	Conceitos básicos de dinâmica .....	18
2.3.1.1	Vibração .....	19
2.3.1.2	Frequência Natural e a teoria de placas .....	19
2.3.1.3	Placas retangulares uniformes isotrópicas .....	19
2.3.1.4	Placas ortotrópicas retangulares .....	20
2.4	Redes Neurais Artificiais .....	21
2.4.1	Classificação das RNA .....	22
2.4.2	Arquitetura de rede .....	24
2.4.3	Vantagens e desvantagens ao se utilizar RNA .....	24
2.5	Trabalhos relacionados .....	25
3	METODOLOGIA .....	28
3.1	Fluxograma do desenvolvimento da RNA .....	28
3.2	Banco de dados .....	29
3.2.1	Caracterização .....	29
3.2.2	Obtenção dos dados .....	29
3.2.3	Filtro de frequência e controle estatístico .....	30
3.2.4	Parâmetros analisados .....	30
3.3	Ambiente de desenvolvimento .....	33
3.3.1	<i>Especificações do dispositivo</i> .....	33
3.3.2	<i>Python</i> .....	33
3.3.3	<i>Anaconda</i> .....	34
3.3.4	<i>Spyder</i> .....	34
3.3.5	<i>Keras</i> .....	34
3.3.6	<i>Sklearn e TensorFlow</i> .....	35
3.4	Arquitetura da rede .....	35
3.4.1	Modelo <i>Sequencial</i> .....	35
3.4.2	Funções de ativação .....	36
3.4.2.1	Sigmoide .....	36
3.4.2.2	<i>Rectified Linear Unit – ReLU</i> .....	36
3.5	Normalização de dados .....	37
3.6	Compilando o modelo .....	37
3.7	<i>EarlyStopping</i> .....	38
3.8	Métrica para avaliação .....	38
3.8.1	Erro médio quadrático (MSE) .....	38
3.8.2	Erro médio absoluto (MAE) .....	39
3.9	Treinamento e validação da rede .....	39
3.10	Predição de rede .....	40
4	RESULTADOS .....	41

4.1	Arquitetura de rede .....	41
4.1.1	Lajes nervuradas .....	41
4.1.2	Lajes pré-moldadas .....	42
4.2	Banco de dados .....	43
4.3	Normalização de dados .....	44
4.4	Definindo parâmetros da arquitetura na rede .....	45
4.5	Métricas de avaliação .....	46
4.5.1	Erro médio quadrático (MSE) e Erro médio absoluto (MAE) .....	46
5	CONCLUSÕES .....	53
	REFERÊNCIAS .....	55
	APÊNDICE A – ALGORITMO DA REDE NEURAL ARTIFICIAL (RNA).....	58



# 1 INTRODUÇÃO

A utilização de materiais mais leves e resistentes, juntamente com o desenvolvimento da tecnologia da construção, tem permitido que se projete estruturas cada vez mais esbeltas, flexíveis e com grandes vãos. Por outro lado, a redução de massa e a diferença no módulo de elasticidade dos elementos afetam diretamente as propriedades dinâmicas da estrutura. A consequência decorrente desse comportamento nos projetos de dimensionamento é a diminuição da frequência natural da estrutura, a qual fica mais próxima da frequência associada às atividades humanas comuns como andar, correr e saltar. Quando submetidas à essas ações, as estruturas se tornam susceptíveis a sofrer vibrações excessivas, interferindo negativamente em critérios de segurança relacionados ao conforto para os usuários, durabilidade da estrutura, aparência e boa utilização de um modo geral (SORIANO, 2014).

Outro ponto importante a se discutir diz respeito à sustentabilidade ligada à essa nova tendência do mercado. Conforme Nagalli (2012), uma edificação sustentável requer o desenvolvimento de novos parâmetros de avaliação, a revisão de procedimentos adotados, o emprego de insumos alternativos, a qualificação da mão de obra, além de outros tantos aspectos. Tem-se, assim, o desenvolvimento de um cenário em que a construção civil como um todo deve buscar novas alternativas para racionalizar as edificações, desde a elaboração de projetos e dimensionamentos de qualidade até o manejo e uso sustentável dos bens materiais e humanos disponíveis à atividade.

Dos elementos que constituem uma estrutura em concreto armado, as lajes, em grande parte dos casos, são submetidas a baixos níveis de tensão quando em utilização. Por esse motivo, no dimensionamento estrutural se atenta apenas às análises desse elemento no estado limite último (ELU), que diz respeito ao estado no qual a estrutura já não pode ser utilizada por razão de esgotamento da capacidade resistente e risco à segurança. Quando uma estrutura é submetida ao ELU, se oferece um risco iminente de ruína, sendo necessários reparos ou até mesmo a substituição da construção para que a segurança seja garantida (SILVA; GASPAR; LOPES, 2014).

Nos projetos de estruturas é necessário analisar as ações externas dinâmicas e determinar o efeito e a relevância das vibrações causadas, para que se possa ter controle, por exemplo, das fissurações excessivas e da forte vibração da estrutura, com base no dimensionamento adequado dos componentes estruturais ou pelo limite determinado pela frequência natural.

A frequência natural é a propriedade que corresponde à taxa na qual um objeto vibra quando é perturbado. Para que se possa mensurar essa propriedade, é necessário seguir algumas das diferentes metodologias disponíveis, tanto de ensaio *in loco* quanto de modelagem computacional.

No caso de estruturas construídas, é possível se obter a frequência natural a partir de testes experimentais simplificados por meio de equipamentos. Já para fase de projetos, se utiliza muito a modelagem computacional para representar a estrutura, inserindo particularidades como propriedades dos materiais e geométricas.

Essas metodologias demandam conhecimento técnico específico, são de elevado custo financeiro em virtude dos equipamentos e softwares, e exigem tempo considerável para execução.

Com o objetivo de desenvolver de uma ferramenta alternativa para mensurar essa propriedade nas estruturas de concreto de forma simplificada, reduzindo o tempo de execução e os custos com os ensaios, propõe-se estudar e elaborar um modelo de previsão de frequências naturais das lajes utilizando Redes Neurais Artificiais (RNA). As técnicas de inteligência computacional, onde se inserem as redes neurais artificiais, visam desenvolver ferramentas capazes de solucionar problemas complexos, de difícil resolução por programas convencionais, de forma rápida e efetiva.

Para viabilizar isso, torna-se necessário apresentar um conjunto de dados ao sistema onde o comportamento estrutural é conhecido, nesse caso composto por uma base de dados própria de leituras *in situ* de lajes pré-moldadas e nervuradas, contendo informações como propriedades geométricas, resistência dos materiais e frequência obtida. Por meio da identificação da influência dos diversos parâmetros na variação da frequência natural das lajes, a rede neural deve ser capaz de prever um novo comportamento, quando inseridos novos parâmetros de entrada.

Essa pesquisa dá continuidade aos estudos desenvolvidos pelo programa de pós graduação em Engenharia Civil e Ambiental da UPF, relativos às análises de vibrações em lajes apresentadas por Lago (2019).

Os resultados obtidos nessa pesquisa têm por objetivo orientar o desenvolvimento de uma ferramenta de fácil uso pelos profissionais quando deparados com condições práticas que necessitem de equipamentos e métodos precisos de avaliação. De posse da análise feita pela rede, pode-se garantir a segurança da estrutura quando submetida à mudança no padrão de ocupação, bem como estimar a frequência natural das lajes de forma simplificada, sem a

necessidade de leitura de sinais obtidos por acelerômetro, resolução analógica de equações simplificadas e/ou utilização de programas computacionais mais complexos.

## **1.1 OBJETIVOS**

### **1.1.1 Objetivo Geral**

O objetivo geral consistiu em usar redes neurais para prever a frequência natural de vibração em lajes nervuradas e pré-moldadas de concreto armado.

### **1.1.2 Objetivos Específicos**

Os objetivos específicos são:

- a) Selecionar a arquitetura de rede para o modelo proposto;
- b) Treinar e avaliar diferentes parâmetros da rede neural;
- c) Verificar a viabilidade da utilização da rede como ferramenta alternativa na obtenção da frequência natural em lajes de concreto armado.

## 2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, com o intuito de delinear o assunto desta pesquisa, busca-se fazer uma breve abordagem sobre os conceitos de sustentabilidade das estruturas, os elementos constituintes e específicos da pesquisa, conceitos de dinâmica estrutural e inteligência artificial (Redes Neurais Artificiais).

### 2.1 Sustentabilidade das estruturas de concreto

A construção e a operação de edifícios são responsáveis por uma parte significativa dos impactos ambientais, ao passo que tornar as cidades mais sustentáveis é uma prioridade global e um dos principais Objetivos de Desenvolvimento Sustentável (ODS) do século.

O Fórum Econômico Mundial (2016) cita que para cada pessoa no planeta, cerca de três toneladas de concreto são fabricadas e utilizadas na construção civil por ano. Considerando todas as etapas de produção, isso representa até 8% de todas as emissões de CO<sub>2</sub> do mundo. Metade dessas emissões são produzidas durante o processo de fabricação do clínquer, material esse que serve como base para a fabricação do cimento, a qual mais consome energia (LEHNE; PRESTON, 2018).

No Brasil, os dados apresentados pelo Sindicato Nacional da Indústria do Cimento (2020) indicam que a produção de cimento no país totalizou 5,7 milhões de toneladas apenas no mês de agosto de 2020, um crescimento de 13,6% em relação ao mesmo mês no ano de 2019.

Além das questões relacionadas a emissão de gases, o consumo do concreto impacta em outros problemas ambientais. Esse material absorve quase um décimo do uso de água industrial do mundo. Fato esse que muitas vezes prejudica o abastecimento humano e a irrigação, visto que 75% do consumo ocorre em regiões com seca e falta de água (MILLER; HORVATH; MONTEIRO, 2018).

Como bem evidenciado por Watts (2019), por centenas de anos a humanidade esteve disposta a aceitar essa desvantagem ambiental em troca dos indiscutíveis benefícios do concreto. Mas a balança agora pode estar se inclinando na outra direção.

Vale lembrar que, apesar dessas preocupações, o concreto é o material “base” do desenvolvimento. Sem ele, muitos dos elementos da vida moderna que consideramos naturais hoje não seriam possíveis – edifícios e casas seguras, estradas, túneis e pontes, acesso a água

potável e energia limpa. O concreto dá abrigo para bilhões de pessoas, fortalece nossas defesas contra desastres naturais e fornece uma estrutura para a saúde, educação, transporte, energia e indústria (GCCA, 2019).

Ainda que haja crescimento na produção e consumo desse material, a indústria do cimento busca se desenvolver em conjunto com a sustentabilidade. O compromisso com a proteção do meio ambiente e com a sociedade se faz presente desde a busca por intensificar o uso de matérias-primas alternativas, melhorar o aproveitamento estrutural na concepção de projeto, reduzir os custos assegurando segurança e durabilidade, até a redução do consumo energético e das emissões de CO<sub>2</sub> (SINDICATO NACIONAL DA INDÚSTRIA DO CIMENTO, 2020).

O ponto interessante é que o desenvolvimento do mercado da construção civil pode ser relacionado a alguns Objetivos do Desenvolvimento Sustentável (ODS), como o ODS 8 (Trabalho decente e crescimento econômico), ODS 9 (Indústria, inovação e infraestrutura) e o ODS 11 (Cidades e comunidades sustentáveis), que foram estabelecidos pela Organização das Nações Unidas (ONU) em 2015, os quais compõem a Agenda 2030, e visam guiar a humanidade ao desenvolvimento sustentável até 2030.

A indústria também se faz importante no papel ambiental. Ao incentivar a pesquisa para o desenvolvimento tecnológico de materiais e projetos, estará contribuindo também com os três pilares da sustentabilidade: econômico, social e ambiental.

Hajek (2019) aponta que a indústria do concreto e de estruturas de concreto podem contribuir para a solução dos ODS. Alguns pontos interessantes de se destacar dizem respeito ao ODS 3 (garantir vida saudável e promover o bem-estar de todos em todas as idades), onde o autor indica que as estruturas em concreto não são tóxicas, não usam produtos químicos perigosos, contribuem para a qualidade acústica das construções, apresentam alta qualidade de desempenho em edifícios e o suporte dado pela infraestrutura pode melhorar a saúde e o bem-estar das pessoas. Com relação ao ODS 9 (Construir infraestrutura resiliente, promover sustentabilidade, industrialização e fomento à inovação), a contribuição da estrutura de concreto é que é possível utilizá-la para o desenvolvimento de materiais de alta qualidade, confiáveis, eficientes em termos de recursos e ambiente construído resiliente (alojamentos e infraestrutura), possível desenvolver e adotar tecnologias limpas e ambientalmente saudáveis nos processos industriais de produção de concreto, elementos e estruturas de concreto, além de que a pesquisa científica é contínua nessa área, o que proporciona a descoberta de tecnologias associadas ao projeto, construção e operação.

É notório a relevância das estruturas de concreto armado no cenário atual, e é importante que o profissional dessa área tenha consciência sustentável sobre isso. Ao realizar projetos, construções e escolha de materiais sustentáveis, estará contribuindo com os ODS, assegurando que a agenda 2030 da ONU seja cumprida e contribuindo com o desenvolvimento sustentável.

Vale lembrar que dentre os elementos da estrutura de concreto, as lajes, por significarem a maior porção do peso próprio das estruturas de concreto armado, ao passo que são submetidas às menores tensões e cerca de 50% do consumo total de concreto da estrutura, são alvo direto do mercado que busca pela redução do consumo de materiais e preza por manter a qualidade da estrutura e segurança dos usuários. Isso justifica a importância de também se estudar dinamicamente esse elemento (PINHEIRO, LIBANIO M., MUZARDO, CASSIANE D., SANTOS, 2010).

## **2.2 Lajes de concreto armado**

As lajes são elementos planos, geralmente horizontais, com duas dimensões muito maiores que a espessura. A função básica estrutural das lajes é receber os carregamentos atuantes, provenientes do peso próprio do elemento e do uso da construção, sejam das atividades humanas, como o caminhar das pessoas, presença de móveis e/ou funcionamento de equipamentos; e transferi-los para os apoios.

Dentre os diversos tipos de lajes, podem-se citar as lajes pré-moldadas e as lajes nervuradas, as quais são alvo de estudo nesta pesquisa.

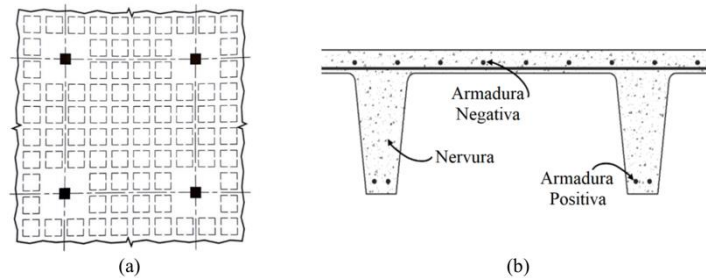
### **2.2.1 Lajes nervuradas**

As lajes nervuradas são elementos constituídos por um conjunto de vigas num sistema monolítico cruzado. A NBR 6118 (2014) conceitua lajes nervuradas como lajes moldadas no local ou com nervuras pré-moldadas, cuja zona de tração para momentos positivos está localizada nas nervuras.

Esse tipo de laje elimina o concreto presente abaixo da linha neutra da seção do elemento, proporcionando a redução do peso próprio e otimizando o aproveitamento do aço e do concreto. Assim como as lajes pré-moldadas, as nervuras são preenchidas por material inerte que tem como única função substituir a massa de concreto, sem colaborar na resistência.

O esquema que representa as lajes nervuradas pode ser observado na Figura 1.

Figura 1 - Sistema de laje nervurada bidirecional: (a) representação em planta e respectivo (b) corte transversal



Fonte: Lago (2019).

### 2.2.2 Lajes pré-moldadas

A laje pré-fabricada unidirecional, também conhecida popularmente como laje pré-moldada, é amplamente utilizada no mercado para edificações residenciais e comerciais com o objetivo de vencer vãos de pequena dimensão.

De acordo com a NBR 14860 (2002), laje pré-fabricada unidirecional é uma laje de seção final maciça ou nervurada, constituída por nervuras principais longitudinais dispostas em uma única direção.

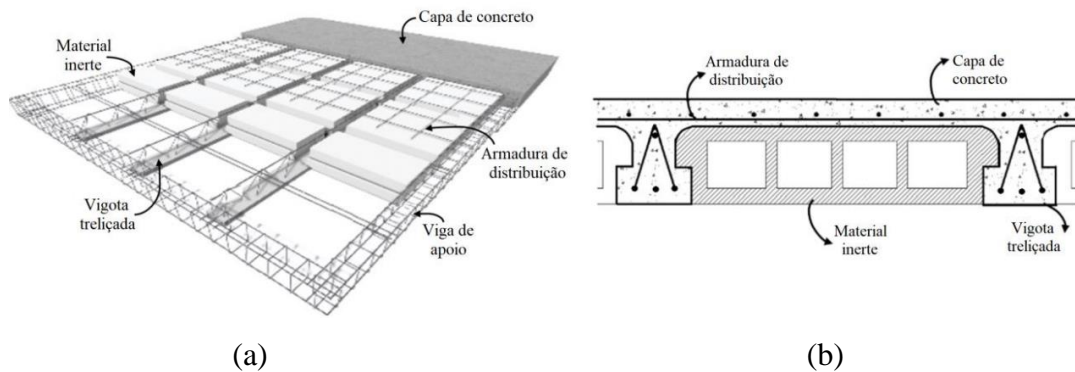
Essas lajes são constituídas basicamente por elementos lineares pré-moldados, chamados de vigotas, fabricados com uma placa de concreto estrutural de espessura de 3 a 5cm, executadas industrialmente fora do local de utilização, dispostas em intervalos regulares.

As vigotas são compostas por uma treliça de aço formada por três banzos paralelos e diagonais laterais de forma senoidal. O aço que compõe o banzo superior das armaduras treliçadas eletrossoldadas da vigota atua como armadura de compressão, principalmente para a montagem e concretagem da laje, resistindo também aos momentos fletores negativos, quando existirem. As barras posicionadas no banzo inferior resistem à tração decorrentes dos momentos fletores positivos.

Entre cada vigota é utilizado um material de preenchimento, esse pré-fabricado de material inerte diverso, podendo sendo maciço ou vazado. Esse material deve ser composto por material leves, suficientemente rígido, que não produza danos ao concreto e às armaduras. Deve ainda ter resistência à ruptura suficiente para resistir à carga de trabalho durante a montagem da laje. Dentre os materiais mais utilizados pode-se citar os blocos vazados de material cerâmico (tabelas) ou concreto e também blocos de poliestireno expandido (EPS). Esses

elementos são desconsiderados como colaboradores nos cálculos de resistência e rigidez da laje. A Figura 2 apresenta a seção transversal para as lajes pré-moldadas

Figura 2 - Sistema de laje pré-moldada treliçada: (a) indicação dos principais elementos que compõem a laje e (b) corte transversal



Fonte: Lago (2019)

### 2.3 Dinâmica das estruturas

Toda estrutura é passível de ações provenientes do comportamento dinâmico externo, causadoras do fenômeno chamado de vibrações. As vibrações podem ser entendidas como movimentos periódicos que se repetem após um intervalo de tempo e que compreendem a energia alternada de energia potencial para energia cinética, ou inversamente. As atividades como tráfego de veículos, impacto de objetos, movimentação de pessoas, máquinas e equipamentos são fontes cotidianas de vibração humana. Outras fontes de vibração também podem ser de origem natural, ocasionadas por fenômenos naturais como ventos e terremotos (RAO, 2008).

Fanella e Mota (2013) enfatizam que muitos critérios relacionados à vibração e ao conforto humano foram propostos ao longo dos anos, no entanto, até a data, não existem padrões que sejam aceitos universalmente. Dessa forma, isso instiga a seguir com os estudos nesse tema, uma vez que o excesso dessas vibrações juntamente com o descaso com a análise dinâmica pode causar desconforto aos usuários e danos às estruturas.

No Brasil, a normativa NBR 6118 (2014) apesar da importância inquestionável do tema, é superficial no tratamento da análise dinâmica e do conforto humano. Segundo ela, a análise das vibrações pode ser feita em regime linear no caso de estruturas usuais. Para estruturas



sujeitas à vibração, deve-se afastar a frequência natural da estrutura ( $f_n$ ) ao máximo possível da frequência crítica ( $f_{crit}$ ), na ordem mínima de 1,20.

O comportamento dessas estruturas pode ser modificado por meio da alteração de alguns fatores, como ações dinâmicas, frequência natural (essa intrinsecamente ligada à rigidez da estrutura ou da massa em vibração) e também o aumento das taxas de amortecimento.

Na falta de se ter valores determinados por métodos experimentais, a normativa orienta a adotar os valores de  $f_{crit}$  indicados na Tabela 1.

Tabela 1 - Frequência crítica para vibrações verticais para alguns casos especiais de estruturas submetidas a vibrações pela ação de pessoas

<b>Caso</b>	<b>Fcrit (Hz)</b>
Ginásio de esportes e academias de ginástica	8,0
Salas de dança ou de concerto sem cadeiras fixas	7,0
Passarelas de pedestres ou ciclistas	4,5
Escritórios	4,0
Salas de concerto com cadeiras fixas	3,50

Fonte: NBR 6118 (2014)

Em casos especiais onde as prescrições não puderem ser atendidas, a norma recomenda uma análise dinâmica mais acurada.

Em muitos sistemas de engenharia, um ser humano age como parte integral do sistema. A transmissão de vibração a seres humanos resulta em desconforto e perda de eficiência. Assim, uma das finalidades importantes de estudar vibrações é reduzi-las por meio do projeto adequado de estruturas, de modo a assegurar que o efeito do desbalanceamento não seja danoso (RAO, 2008).

### 2.3.1 Conceitos básicos de dinâmica

Para que se possa compreender melhor sobre o assunto a ser tratado, no decorrer desse tópico serão apresentados alguns conceitos básicos sobre dinâmica das estruturas, como vibrações, frequências naturais e teoria das placas, os quais serão de suma importância para o entendimento do desenvolvimento da pesquisa.

### 2.3.1.1 Vibração

Qualquer movimento que se repita após um intervalo de tempo é denominado vibração. A teoria de vibração trata do estudo de movimentos oscilatórios de corpos e forças associadas a eles. A vibração pode ser classificada de diversas maneiras, sendo as mais importantes:

a) Vibração livre: acontece após a perturbação inicial de um sistema, onde ele continua a vibrar por conta própria e nenhuma outra força externa é atuante. A oscilação de um pêndulo é um exemplo desse tipo de vibração. O sistema em vibração livre tenderá a vibrar em certa frequência predominante, a chamada frequência natural de um sistema (SILVA, 2006).

b) Vibração forçada: quando o sistema está sujeito a uma força externa. Essa oscilação surge comumente em estruturas.

### 2.3.1.2 Frequência Natural e a teoria de placas

Frequência natural, ou também conhecida como frequência de ressonância, é a frequência (ou conjunto de frequências) particular de um corpo em vibração livre, determinada pelo tamanho, forma e composição desse. Sempre que a frequência da força externa coincide com uma das frequências naturais do sistema, ocorre uma condição conhecida como ressonância, que resulta em deflexões excessivas e falha estrutural. Nesse caso, uma vibração comparavelmente fraca, em ressonância, pode produzir vibrações mais intensas, pois o corpo recebe energia da fonte externa periodicamente.

A teoria de placas é muito empregada para descrever modelos de análise de vibrações. Sabe-se que um elemento de placa trabalha predominantemente à flexão, curvando-se perpendicularmente ao seu próprio plano (BLEVINS, 2016).

Na teoria de flexão em placas, o modelo de placas de *Kirchhoff* e o modelo de placas de *Mindlin* são os mais utilizados para análise. O primeiro é comum em casos de placas finas em que a deformação por cisalhamento é negligenciada. Já as placas de *Mindlin* são utilizadas quando esse efeito não pode ser desprezado como, por exemplo, quando a placa é relativamente espessa em relação às suas outras dimensões.

### 2.3.1.3 Placas retangulares uniformes isotrópicas

A teoria das placas finas com pequenas deflexões, ou também chamada de teoria de *Kirchhoff* é utilizada para se obter a equação do movimento não amortecido para pequenas

deformações ( $w$ ) em placas uniformes e elásticas no plano cartesiano  $x - y$ , negligenciando as ações no plano normal à placa.

A frequência natural em placas pode ser expressa em termos de um parâmetro adimensional  $\lambda$ , da massa por unidade de área da placa  $\gamma$  e do módulo de Elasticidade  $E$ , conforme Equação (1), dada por Blevins (2016).

$$f_{ij} = \frac{\lambda_{ij}^2}{2\pi \cdot a^2} \sqrt{\frac{E \cdot h^3}{12 \cdot \lambda(1 - \nu^2)}} \quad (1)$$

Fica claro que a frequência das placas aumenta com o aumento da sua espessura e diminui com o aumento do comprimento  $a$ . O parâmetro adimensional  $\lambda$  é função das condições de apoio, da geometria da placa, e do coeficiente de *Poisson* para bordas livre e bordas curvas simplesmente apoiadas.

Pela suposição da isotropia nas placas, as propriedades do material que as compõem são as mesmas em todas as direções, o que implica a ocorrência de deformações principalmente no sentido do carregamento aplicado. As deformações dependem de duas constantes elásticas independentes, por exemplo  $E$  e  $\nu$ .

#### 2.3.1.4 Placas ortotrópicas retangulares

Algumas placas são constituídas de materiais que possuem propriedades que variam com a direção considerada, os chamados materiais anisotrópicos. Em sistemas anisotrópicos submetidos a certo carregamento, as deformações nas demais direções podem ser maiores ou menores do que a deformação no sentido do carregamento aplicado, dependendo das propriedades.

A fórmula que descreve de maneira generalizada a frequência natural para diferentes geometrias de placas ortotrópicas, conforme Blevins (2016), é dada pela Equação (2).

$$f_{ij} = \frac{\pi}{2\gamma^{1/2}} \left[ \frac{G_i^4 D_x}{a^4} + \frac{G_i^4 D_x}{b^4} + \frac{2[H_i H_j D_{xy} + 2D_k (J_i J_i - H_i H_j)]}{a^2 b^2} \right]^{1/2} \quad (2)$$

Da equação anterior,  $G$ ,  $H$  e  $J$  são determinados em função dos índices  $i$  e  $j$ , e das condições de apoio das placas. Já os parâmetros de rigidez ( $D_x$ ,  $D_y$ ,  $D_{xy}$  e  $D_k$ ) dependem da sua geometria.

## 2.4 Redes Neurais Artificiais

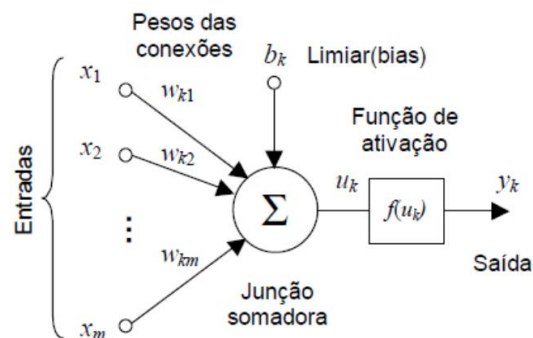
As redes neurais artificiais (RNA) são sistemas muito semelhantes ao sistema nervoso humano, o qual trabalha recebendo dados, processando-os, entendendo-os e fazendo uma decisão racional a respeito da informação recebida; e então armazena o conhecimento dessa experiência para decisões futuras. As RNA estão se tornando fortes ferramentas para muitos problemas de engenharia. Elas podem realizar classificações de acordo com os dados de entrada e gerar previsões de saída (AMIT BANERJEE, G. POHIT, 2016).

As RNA são particularmente úteis quando se está lidando com situações nas quais o estabelecimento de relações funcionais entre variáveis de entrada e de saída são excessivamente complexas ou não estão disponíveis (SILVA; GOMES, 2016).

O sucesso no processo de aprendizagem da rede está diretamente ligado à arquitetura da rede – quantidade de camadas ocultas e número de nós dessas camadas – e passa igualmente pelas estratégias utilizadas no treinamento.

Em um modelo geral, um neurônio artificial pode ser representado como no esquema da Figura 3 a seguir:

Figura 3 - Neurônio artificial genérico



Fonte: Zuben (1996)

A partir da estrutura e funcionamento de um neurônio biológico é criado o neurônio artificial genérico (ou matemático). O neurônio artificial é um modelo simplificado do neurônio biológico, inspirado a partir da análise da geração e propagação de impulsos elétricos pela membrana celular dos neurônios. Esse neurônio recebe um ou mais sinais de entrada e devolve um único sinal de saída, que pode ser distribuído como sinal de saída da rede ou como sinal de entrada para um ou mais neurônios da camada posterior (que formam as redes neurais).

As ramificações do neurônio biológico, chamadas de dendritos e axônios, são representados matematicamente apenas pelas sinapses, e a intensidade da ligação é representada por uma grandeza chamada peso sináptico. Quando as entradas são apresentadas ao neurônio, elas são multiplicadas pelos pesos sinápticos correspondentes, gerando as entradas ponderadas. Isso descreve uma das bases matemáticas fundamentais do funcionamento da rede neural: a multiplicação de matrizes.

O neurônio então totaliza todas as multiplicações gerando um único resultado. A essa função denomina-se junção somatória ou função de combinação  $\Sigma$ . Esse valor é apresentado à uma função de ativação que tem a finalidade de evitar o acréscimo progressivo dos valores de saída ao longo das camadas da rede, visto que essas funções possuem valores máximos e mínimos contidos em intervalos determinados. A ativação do neurônio depende da aplicação da função de ativação, uma vez que se a soma dos impulsos recebidos ultrapassa o limiar de excitação estabelecido pela função, o neurônio é ativado e “disparado”.

No neurônio biológico, os sinais são enviados em pulsos e alguns componentes funcionam como filtros de frequência, a exemplo do axônio. O modelo artificial também pode incluir uma polarização, conhecida como bias de entrada. Essa variável é incluída no somatório da função de ativação com o intuito de aumentar o grau de liberdade dessa função e a capacidade de aproximação da rede. O *bias* possibilita que um neurônio apresente uma saída não nula ainda que todas as entradas fossem nulas. (DEEP LEARNING BOOK, 2021)

### 2.4.1 Classificação das RNA

A definição da quantidade de neurônios e as camadas ocultas presentes na rede neural é uma das tarefas mais difíceis do projeto de uma rede. A arquitetura de uma rede neural é criada a partir da definição do número de camadas e a quantidade de nós que deve integrar cada camada. As camadas que compõem uma rede neural podem ser classificadas como:

a) Camada de entrada: camada da rede que recebe os dados a serem apresentados à rede para o processamento. A quantidade de nós (ou neurônios) existente nessa camada representa o número de variáveis de entrada que determinado problema possui;

b) Camadas ocultas: nesse trecho são encontradas as unidades ocultas da rede, que são unidades de processamento, onde são realizadas as operações mais significativas das redes neurais. São chamadas de ocultas pois são internas à rede, e as saídas não têm relação nessa camada;

c) Camada de saída: onde contém as respostas do problema. A quantidade de neurônios existentes nessa camada dependerá da quantidade de saídas desejadas pelo problema.

No que se refere aos tipos de redes existentes, podem ser classificadas da seguinte maneira:

a) Redes diretas ou acíclicas (*feed-forward networks*): são as quais todas as conexões apontam uma única direção – da camada de entrada à camada de saída. Essa tipologia é comumente encontrada nos *perceptrons* multicamadas, o qual pode ser visto como o tipo de arquitetura mais simples de rede *feed-forward*;

b) Redes recorrentes ou cíclicas (*recurrent networks*): aquelas cuja conectividade possui pelo menos um laço de realimentação, onde a saída de algum neurônio é também usada como entrada.

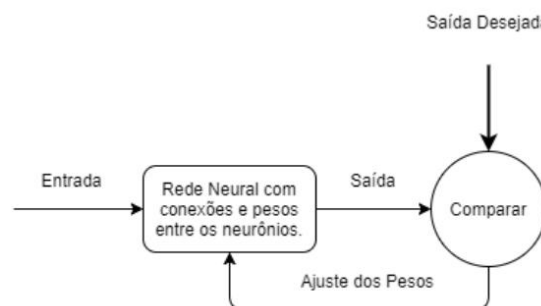
Se analisado o número de camadas existentes, a rede neural também pode ser classificada em dois grupos:

a) Redes de camada única (*single layered perceptron*): onde possuem apenas uma camada além da camada de entrada. Vale salientar que a camada de entrada não entra para a contagem nessa classificação por não ocorrer nenhum processamento;

b) Redes de múltiplas camadas (*multilayer perceptron*): as quais possuem mais de uma camada além daquela de entrada.

Na Figura 4 é representada uma rede neural com três camadas de neurônios conhecida como *feed forward network*. Esse tipo de rede tem se tornado muito popular porque há uma razoável disponibilidade de métodos de aprendizado disponíveis, como o de retro-propagação (*backpropagation*).

Figura 4 - Funcionamento da RNA na fase de treinamento



Fonte: Costa (2019)

A *Backpropagation Neural Network* (BPNN) é composta por três camadas: a camada de entrada, a oculta e a de saída. Cada uma é formada por neurônios que se conectam por meio

de pesos. Cada neurônio possui um peso denominado *bias* e uma função de ativação. As funções de ativação decidirão se um neurônio será ou não ativado.

### 2.4.2 Arquitetura de rede

Não há procedimento teórico para se determinar a arquitetura ideal de rede, uma vez que cada rede possui um banco de dados e uma finalidade de resolução de problema específico.

Estudos anteriores orientam que redes que apresentam uma única camada oculta podem aproximar relação funcional para determinado nível de precisão desejado, na qual o número de neurônios utilizados é usualmente determinado por tentativa e erro.

Flood (1999) sugeriu que o número de neurônios na camada oculta deve estar entre a soma e a média do número de neurônios das camadas de entrada e de saída da rede neural (*apud* CAVALCANTI, 2010).

Alguns procedimentos para o desenvolvimento de uma rede neural artificial são sintetizados a seguir (PARKER, D. B. 1982 *apud* CAVALCANTI, 2010):

- a) um maior número de elementos de processamento pode levar a rede a um processo de memorização, resultando numa situação que não é desejável;
- b) quanto mais complexa a função de mapeamento (relação entre as entradas e saída da rede) mais complexa deve ser a RNA;
- c) quanto maior for o conjunto de treinamento, maior deverá ser o número de elementos de processamento da RNA (neurônios).

### 2.4.3 Vantagens e desvantagens ao se utilizar RNA

Existem diversos fatores determinantes na hora de se optar pela utilização de redes neurais para se resolver problemas. A capacidade de aprendizagem, por meio da generalização, abstração e extração de propriedades estatísticas dos dados de entrada é um dos pontos positivos das redes. Aliada à capacidade de processamento eficiente, tolerância de falhas, flexibilidade de implementação e não linearidade, constituem as habilidades mais importantes que as redes neurais possuem. Por outro lado, deve-se atentar que as redes neurais precisam de treinamento para que funcionem. Demandam elevado tempo de trabalho com a escolha das arquiteturas (geralmente grandes e complexas), e não existem regras claras e objetivas para se definir a melhor configuração para problemas específicos.

Como declarado por Parker (1982), uma rede neural artificial é um processador paralelo distribuído que tem a propensão natural para armazenar conhecimento e torná-lo disponível para uso.

Uma rede armazena conhecimento nos pesos, que são atribuídos a cada conexão entre os neurônios. Esses pesos são conhecidos como pesos sinápticos, que podem ser números reais ou inteiros, que serão ajustados durante o processo de treinamento.

## 2.5 Trabalhos relacionados

Diversos estudos têm sido conduzidos utilizando redes neurais artificiais para resolução de problemas de engenharia envolvendo dinâmica estrutural. Fonseca (2003) propôs um sistema de avaliação do comportamento estrutural de vigas por meio de uma identificação da influência dos diversos parâmetros na carga e nos tipos de comportamentos resultantes (plastificação, enrugamento e flambagem global), estabelecendo limites mais flexíveis entre cada um. A análise foi conduzida empregando um sistema híbrido de redes neurais e de lógica nebulosa, apresentando os dados de treinamento e otimizando fórmulas de projeto existentes com algoritmos genéricos.

Sahin e Sheno (2003) realizaram um estudo numérico e experimental a fim de localizar e quantificar danos em uma viga de aço. Para o estudo, foi utilizada a técnica de RNA tendo, na etapa do seu treinamento, os modos e as frequências obtidas numericamente, por meio do software ANSYS. Buscando o melhor desempenho da RNA, os autores criaram três diferentes redes, cada uma com um tipo de dado de entrada, dentre os quais estavam as frequências, diferença máxima entre as curvaturas e os modos de vibração. O treinamento foi realizado com dados de 480 cenários construídos com severidades e posições de danos diferentes. Para validar a técnica, a RNA foi testada com os dados obtidos do modelo experimental e os resultados obtidos permitiram concluir que as diferenças percentuais entre as frequências são dados que podem indicar a existência e a severidade do dano, no entanto, em se tratando de localização, a diferença máxima entre as curvaturas apresentou resultados mais precisos.

Herbert, H. M, (2004) apresenta conceitos e princípios para a utilização de Redes Neurais Artificiais para avaliação e detecção de danos em estruturas. O autor mostra interessantes aplicações dessa técnica e acrescenta que o uso de redes neurais se mostrou bastante promissor, indicando um grande potencial para utilização.

Brito Júnior *et al.*, (2007) aplicaram o algoritmo Levenberg-Marquardt para prever o comportamento dinâmico-mecânico de compósitos de matriz epóxi reforçados com fibras de



carbono. Uma rede neural do tipo “perceptron” de múltiplas camadas foi empregada, e os resultados mostraram que a aplicação do algoritmo de aprendizado Levenberg-Marquardt conduz a uma elevada qualidade preditiva para compósitos de matriz epóxi, uma vez que o coeficiente de desempenho se apresentou superior a 0.9 para 64% dos conjuntos testados. Os testes iniciais consideraram uma arquitetura simples 2-[30-30]2-1, resultando em uma baixa qualidade preditiva. Entretanto, o aumento do número de neurônios nas camadas escondidas resultou em uma arquitetura otimizada 2-[100-100]2-1.

Mehrjoo *et al.* (2008) apresentaram um método utilizando redes neurais com algoritmo *Backpropagation*, capaz de estimar a intensidade de danos presentes em articulações de pontes treliçadas. Por se tratar de grandes estruturas com muitas articulações, conseqüentemente com muitos parâmetros, o método propõe uma divisão da estrutura em subestruturas que possam representar a ponte completa. A rede recebeu como dados de entrada as frequências e os modos naturais de vibração das estruturas testadas. O treinamento da rede foi realizado com os dados obtidos de uma subestrutura, e a partir do seu aprendizado pôde-se generalizar a referida rede, a fim de conhecer os dados de saída, ou seja, os danos da estrutura completa. Os resultados indicaram um bom aprendizado da rede mostrando, portanto, boa precisão na identificação dos danos.

Karimi *et al.* (2013) utilizaram as frequências naturais de vigas, modeladas numericamente, para alimentar redes neurais na localização de danos. A rede foi treinada com algoritmos genéticos, e teve como alimentação e resposta os valores das frequências naturais, a previsão da localização e a profundidade das trincas, respectivamente. Após os resultados, os autores concluíram que a técnica apresentou bons resultados, com erros em torno de 4% do esperado.

Kondru e Rao (2013) utilizaram a estrutura de uma viga em balanço para aplicar a técnica de RNAs com vistas à localização e mensuração da profundidade de trincas. Nesse trabalho, a rede foi treinada com frequências naturais da estrutura e foi comparada sua performance na localização das trincas com os resultados obtidos por meio do software ANSYS.

Oliveira *et al.*, (2016) utilizaram a técnica das Redes Neurais Artificiais para o dimensionamento de pilares de concreto armado submetidos à flexão oblíqua composta. Foram estudadas seções transversais retangulares e circulares maciças. A rede serviu para a obtenção de um mapeamento das variáveis de projeto em pilares de concreto. O treinamento foi baseado na retropropagação do erro, com a consideração de mais de 400 dados para cada tipo de seção

transversal estudada. Os resultados obtidos indicaram um excelente desempenho da técnica em situações usuais de projeto.

Ahmed (2016) em sua tese de doutorado reservou parte da sua pesquisa para avaliar o desempenho das Redes Neurais Artificiais, especificamente a rede *Backpropagation*, na estimativa dos tamanhos dos vazios em lajes de concreto armado. Para a análise, foram desenvolvidos modelos numéricos no ANSYS, dos quais foram obtidas suas 10 primeiras frequências, a fim de obter as informações necessárias para treinar a RNA. Uma vez que a rede apresentou um bom desempenho, isto é, o erro durante a fase de aprendizado da rede estava por volta de 0,0004, passou-se para a segunda etapa, a fase de testes. Deste modo, foram apresentadas as 10 primeiras frequências da estrutura real e, ao final do processamento, observou-se que as regressões dos resultados dos testes mostraram que a RNA proposta foi capaz de prever os tamanhos de vazios com uma precisão consideravelmente alta. É importante ressaltar que o autor chama a atenção para o número de neurônios de cada simulação, uma vez que foi observado que o aumento das células artificiais melhorou significativamente o poder de generalização da rede neural. Ainda como conclusões da análise, Ahmed (2016) classifica a técnica como uma ferramenta eficiente para detectar o tamanho de vazio em lajes.

Abambres e Lantsoght (2020) propuseram uma fórmula baseada em redes neurais artificiais (RNA) para obter estimativas da capacidade de cisalhamento de lajes de concreto armado unidirecional sob uma carga concentrada, com base em 287 resultados de testes obtidos na literatura. Foi realizado o desenvolvimento do banco de dados com resultados experimentais da literatura, e também o desenvolvimento da formulação de matriz baseada em RNA. Com o estudo, pode-se observar que o modelo “ótimo” proposto produz erros relativos máximos e médios de 0,0% para os 287 pontos de dados.

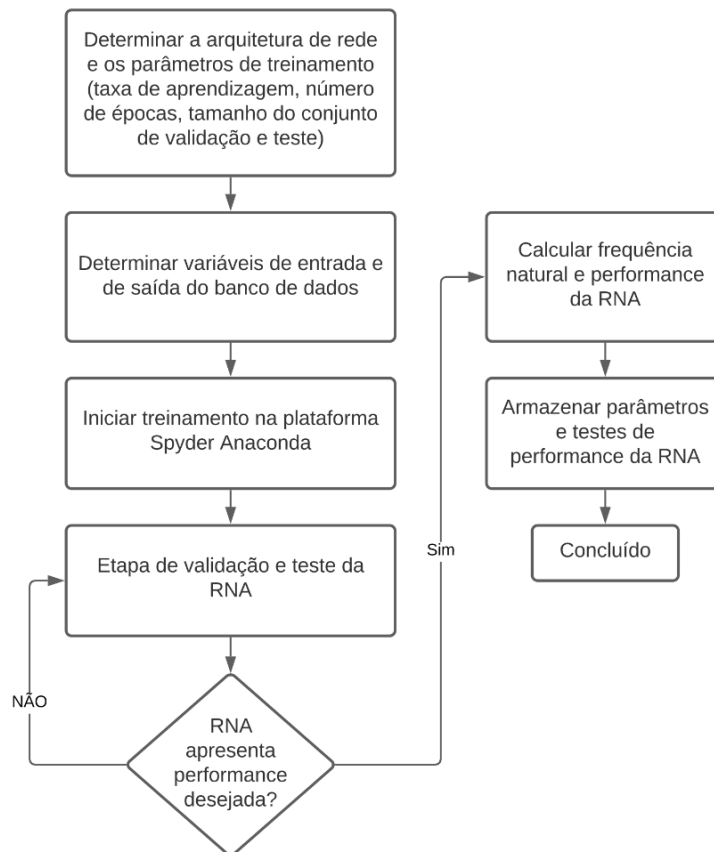
### 3 METODOLOGIA

Neste tópico é apresentada a metodologia utilizada para o desenvolvimento da rede neural artificial.

#### 3.1 Fluxograma do desenvolvimento da RNA

Um fluxograma do desenvolvimento da RNA é apresentado na Figura 5.

Figura 5 - Fluxograma de desenvolvimento da RNA



Fonte: O autor (2022)

Conforme os critérios sugeridos pelos autores estudados, inicialmente determina-se uma arquitetura de rede que seja compatível com o problema de engenharia a ser resolvido. Nessa etapa, é também importante estimar alguns parâmetros como: taxa de aprendizagem, número de épocas, tamanho do conjunto teste e do conjunto de validação que irá se utilizar na rede neural.

Após avaliar as arquiteturas e decidir qual será a mais adequada, e com posse do banco de dados devidamente formatado, irá se definir a quantidade de dados pertencente ao conjunto teste e ao conjunto validação. Nesse estágio, pode-se avaliar se a quantidade de dados é suficiente para iniciar o treinamento ou se será necessário ampliar o banco de dados com novas leituras *in loco*, ou ainda gerar novos dados com as equações simplificadas.

## **3.2 Banco de dados**

### **3.2.1 Caracterização**

O banco de dados é resultado de um estudo realizado por Lago (2019), dissertação pela qual se dá continuidade nesta pesquisa. O banco é composto por 26 lajes nervuradas e 26 pré-moldadas, de três diferentes construções, as quais foram selecionadas para realizar os ensaios experimentais e os cálculos analíticos. Todas os pisos apresentavam revestimento cerâmico e os espaços entre as nervuras eram preenchidos com Poliestireno Expandido (EPS).

### **3.2.2 Obtenção dos dados**

As características dinâmicas foram obtidas com o auxílio de um acelerômetro, baseado em sistemas Microeletromecânicos (MEMS) da marca SENSR, modelo GP2Lx. O equipamento é fixado na estrutura e o cabo USB conectado ao computador, por meio do qual são transmitidas as respostas de vibração no domínio do tempo. O sistema funciona em conjunto com um software, o qual permite visualizar, gravar, analisar e exportar as respostas observadas em tempo real.

Com relação ao ensaio de vibração livre, foi executado pelo ensaio de impacto de calcanhares, onde se obtém a aceleração no domínio do tempo. Com os dados de aceleração são obtidas as frequências naturais pela transformada de *Fourier*.

Os ensaios foram realizados por uma pessoa de aproximadamente 55 kg, com estatura de 1,70 m. O tempo de extração dos sinais de aceleração iniciou-se assim que os estímulos cessaram e foi finalizado segundos após a aplicação do carregamento dinâmico.

Após a identificação das frequências naturais, foram obtidos os fatores de amortecimentos por meio do método do decremento logarítmico e da largura da banda, utilizando sinais provenientes do acelerômetro.

### 3.2.3 Filtro de frequência e controle estatístico

Para que se pudesse separar o sinal original da estrutura do ruído provocado pelos fatores externos, e também reduzir os erros de amostragem, foi aplicado um filtro de frequência aos sinais obtidos.

Os autores elaboraram uma rotina no programa *Scilab-6.0.1* com a utilização de um filtro passa-baixa, do tipo *Butterworth* de quarta ordem. Esse filtro é popularmente utilizado para obter sinais abaixo de uma certa frequência, pois é projetado para produzir respostas mais planas possíveis até a frequência de corte. A utilização do filtro possibilitou selecionar o harmônico de interesse e alcançar as curvas mais suaves no domínio do tempo para a obtenção apurada do amortecimento pelo método do decremento logarítmico (LAGO, 2019).

A utilização do filtro de frequência também foi útil na aplicação do método da largura da banda, uma vez que algumas lajes possuem picos de frequência muito próximos uns dos outros, o que pode aumentar a largura da banda e distorcer os valores de interesse. Quanto às frequências naturais, houve pouca ou nenhuma alteração dos valores obtidos antes da aplicação do filtro.

No que diz respeito ao controle estatístico dos dados obtidos pelo acelerômetro, foram calculados os coeficientes de variação dos resultados e comparados com as faixas de precisão (Tabela 2) propostas por Gomes (2009), as quais avaliam a confiabilidade dos resultados experimentais.

Tabela 2 - Classificação do índice de precisão de experimentos conforme avaliação do coeficiente

C.V.	Avaliação do C.V.	Precisão
< 10%	Baixo	Alta
10 a 20%	Médio	Média
20 a 30%	Alto	Baixa
> 30%	Muito alto	Muito baixa

Fonte: GOMES (2009)

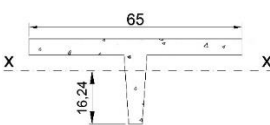
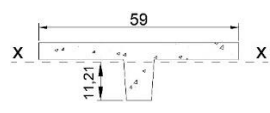
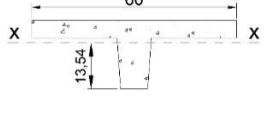
### 3.2.4 Parâmetros analisados

As lajes nervuradas possuem nervuras com altura  $h_t$  nas duas direções, sobre as quais há uma camada de concreto com espessura  $h_c$ . O espaçamento entre os eixos das nervuras é designado por  $S$ , a largura das nervuras é representada por  $b$ , e  $h$  é a altura final da laje. Como

todas as lajes nervuradas são simétricas nas direções  $x$  e  $y$ , as dimensões  $S$ ,  $b$  e  $L$  são as mesmas nos dois sentidos (Tabela 3).


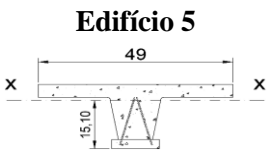

As lajes pré-moldadas treliçadas são distribuídas em uma direção (geralmente na direção do vão menor,  $B$ ) e concretadas monoliticamente com uma capa de concreto superior às nervuras com espessura  $h_c$ . Na condição de apoio, S representa borda simplesmente apoiada e C borda engastada. O espaçamento entre os eixos das nervuras é dado por  $S$ , a largura das nervuras é representada por  $b$ , e  $h$  é a altura final da laje (Tabela 4). Para a composição do banco de dados, a combinação das vinculações de cada laje foi identificada em uma coluna adicional nas tabelas, convertidas em valores de 1 a 6 (Tabela 5).

Tabela 3 - Resumo das características geométricas das lajes nervuradas (dimensões em cm)

	Laje	Condição de apoio	$L$	$B$	$S$	$b$	$h_l$	$h_c$	$h$	$h_e = (12I/S)^{1/3}$	Frequência Fn - acelerômetro Hz
<b>Edifício 1</b> 	LN01	S-C-S-S	633,0	303,0							28,52
	LN02	S-C-C-C	888,5	488,0							25,78
	LN03	S-C-S-C	1090,5	363,0							32,29
	LN04	S-C-S-C	701,5	491,5							31,64
	LN05	S-C-C-C	671,5	369,5							39,19
	LN06	S-C-S-C	861,0	303,0	65,0	7,0	21,0	5,0	26,0	16,24	31,70
	LN07	C-C-S-S	1004,5	488,0							24,22
	LN08	S-C-S-S	1004,5	382,5							29,95
	LN09	S-C-S-C	421,0	353,0							44,53
	LN10	S-C-C-C	639,5	419,0							32,42
<b>Edifício 2</b> 	LN11	S-C-S-C	408,0	363,0							31,25
	LN12	S-C-S-C	408,0	318,0							41,80
	LN13	S-C-C-C	339,0	188,0							71,35
	LN14	C-C-C-C	353,0	338,0							38,15
	LN15	S-C-C-C	473,0	353,0							47,40
	LN16	S-C-S-S	338,0	210,0	59,0	9,0	12,0	5,0	17,0	11,21	56,77
	LN17	S-C-S-S	218,0	128,0							117,02
	LN18	S-C-S-C	408,0	318,0							49,48
	LN19	C-C-C-C	353,0	338,0							43,23
	LN20	S-C-S-C	353,0	201,5							47,40
	LN21	S-C-S-S	338,0	218,0							67,71
<b>Edifício 3</b> 	LN22	S-C-S-S	1252,0	805,0							7,82
	LN23	S-C-S-C	965,0	805,0							7,82
	LN24	S-C-S-S	960,0	805,0	60,0	10,0	15,0	5,0	20,0	13,54	7,04
	LN25	S-C-S-S	965,0	805,0							7,84
	LN26	S-C-S-C	960,0	805,0							7,93

Fonte: Adaptado de LAGO (2019).

Tabela 4 - Resumo das características geométricas das lajes pré-moldadas treliçadas (dimensões em cm)

	Laje	Condição de apoio	$L$	$B$	$S$	$h_t$	$h_c$	$h$	$h_e = (12l/S)^{1/3}$	Frequência Fn - acelerômetro Hz
 <p><b>Edifício 4</b></p>	LP01	S-C-S-S	517,0	331,0						36,92
	LP02	S-C-S-S	266,0	246,0						47,14
	LP03	S-C-S-S	317,0	295,0						59,38
	LP04	S-C-S-S	310,0	185,0	42,0	8,0	4,0	12,0	9,43	36,46
	LP05	C-C-S-S	326,5	256,0						36,46
	LP06	C-C-S-S	404,0	326,5						30,15
	LP07	S-C-S-S	358,5	336,0						65,10
 <p><b>Edifício 5</b></p>	LP08	S-C-S-S	696,0	527,5						16,62
	LP09	C-C-S-S	716,0	665,0						14,58
	LP10	S-C-S-S	696,0	527,5						15,36
	LP11	C-C-S-S	866,0	432,5	49,0	16,0	4,0	20,0	15,10	21,61
	LP12	S-C-S-S	671,0	562,5						17,97
	LP13	S-C-S-S	671,0	562,5						17,06
	LP14	S-C-S-S	921,0	452,5						14,84
	LP15	C-C-S-S	866,0	867,5	49,0	20,0	5,0	25,0	18,87	14,64
	LP16	S-C-S-S	921,0	452,5						23,05
	LP17	S-C-S-S	889,0	457,0						33,75
	LP18	C-C-S-S	389,5	386,0						25,52
	LP19	C-C-S-S	482,5	519,5	49,0	16,0	4,0	20,0	15,10	34,77
LP20	C-C-S-S	540,0	333,0						35,94	
LP21	C-C-S-S	543,5	540,0						31,51	
LP22	C-C-S-S	500,5	305,5						57,94	
 <p><b>Edifício 6</b></p>	LP23	S-C-S-S	340,0	196,0	49,0	16,0	5,0	21,0	16,45	60,94
	LP24	S-S-S-S	434,0	389,0						28,26
	LP25	S-S-S-S	200,0	191,0	49,0	8,0	5,0	13,0	10,22	95,24
	LP26	S-S-S-S	230,0	121,0						62,62

Fonte: Adaptado de LAGO (2019).

Tabela 5 - Relação de condições de apoio e banco de dados

Condição de apoio	Número representado no banco de dados
S-C-S-S	1
S-C-C-C	2
S-C-S-C	3
C-C-S-S	4
C-C-C-C	5
S-S-S-S	6

Fonte: O autor (2022)

Foram incluídos ao banco de dados os valores de frequência natural obtidos pelo acelerômetro. Após, o banco de dados foi formatado em arquivo CSV (*comma-separated values*) para que o software pudesse interpretar. Basicamente, um arquivo CSV contém informações na forma de tabela com os dados delimitados por vírgulas (ou algum outro tipo de caractere separador) e os registros separados por quebras de linha.

### 3.3 Ambiente de desenvolvimento

#### 3.3.1 Especificações do dispositivo

O dispositivo utilizado para o desenvolvimento da rede possui processador Intel (R) Core (TM) i7-9700 CPU @ 3.00GHz, com 16,0 GB de memória RAM instalada, no sistema operacional de 64 bits. O tempo de execução para cada fase de treinamento durou em torno de 4 minutos por teste.

#### 3.3.2 Python

A CPU (*Central Processing Unit*), ou também Unidade Central de Processamento, é onde todas as operações do computador são processadas. Ela interpreta uma linguagem conhecida como *linguagem de máquina*, a qual é relativamente simples, uma vez que é representada apenas por zeros e uns, porém que demanda muito trabalho para se escrever.

Por esse motivo, se utilizam vários tradutores de linguagem, como *Python* e *JavaScript*, por exemplo. Esses tradutores são capazes de converter os códigos gerados para linguagem de máquina, possibilitando serem executados pela CPU.



*Python* é uma linguagem de alto nível utilizada para o aprendizado de máquina e análise de dados. Um dos seus maiores atrativos é possuir um grande número de bibliotecas, nativas e de terceiros, o que faz com que seja difundida e útil em uma grande variedade de setores dentro do desenvolvimento web (GONZÁLEZ DUQUE, 2000).

Uma das características interessantes das linguagens de alto nível é que os programas escritos nesse formato podem ser executados entre diferentes computadores usando um diferente interpretador ou recompilando o código.

### 3.3.3 *Anaconda*

Existem diversas opções de ferramentas e plataformas para o desenvolvimento em *Python*. A plataforma mais popular, e também a utilizada neste estudo, é a *Anaconda*. A *Anaconda* é uma ferramenta computacional gratuita e de fácil instalação, a qual permite gerir distribuições em linguagem *Python*, ambientes de trabalho e módulos. Ela contém mais de 720 pacotes no formato *open-source* e é possível de se utilizar em sistemas operacionais *Windows*, *Mac OSX* e *Linux* (ANACONDA, 2021).

### 3.3.4 *Spyder*

Dentre esses pacotes, encontra-se o *Spyder*, o qual é uma ferramenta de interface gráfica, semelhante ao *Matlab*, que permite a utilização de linguagem *Python* num ambiente interativo. Além disso, o *Spyder* permite a instalação de vários módulos científicos, como o *Numby* (álgebra linear), *Matplotlib* (plotagem 2D e 3D) e *Scipy* (processamento de imagens) (SPYDER, 2021).

### 3.3.5 *Keras*

O *Keras* é uma biblioteca de rede neural de código aberto escrita em *Python*. Projetado para permitir experimentação rápida com redes neurais profundas, ele se concentra em ser fácil de usar, modular e extensível.

O *Keras* contém várias funções para construir partes importantes de redes neurais, como camadas, funções de perda, funções de ativação, otimizadores, entre outras. O código está hospedado no *GitHub*, o qual é um serviço baseado em nuvem que hospeda um sistema de controle de versão chamado *Git*. Ele permite que os desenvolvedores colaborem e façam

mudanças em projetos compartilhados enquanto mantêm um registro detalhado do progresso (KERAS, 2021).

Além das redes neurais padrão, *Keras* tem suporte para redes neurais convolucionais e recorrentes. *Keras* também permite que os usuários produzam modelos profundos em smartphones (iOS e Android), na web ou na Java Virtual Machine.

### 3.3.6 *Sklearn e TensorFlow*

O *TensorFlow* é uma biblioteca de baixo nível, utilizado para construir modelos de aprendizado de máquina com conjunto de operadores simples. Já o *scikit-learn* (ou apenas *sklearn*) é uma biblioteca de aprendizado de máquina de código aberto e de nível superior, utilizada para a linguagem de programação *Python*. Ela possui algoritmos de classificação, regressão e agrupamento que servem para simplificar a programação aos desenvolvedores.

## 3.4 Arquitetura da rede

O algoritmo da rede neural artificial seguiu o modelo *perceptron* de multicamadas. Essa forma de rede consiste numa estrutura dividida entre camadas de neurônios de entrada (representados pelas variáveis de entrada), neurônios intermediários (responsável por realizar as interações entre os parâmetros de entrada e de saída e pela primeira aplicação da função de ativação) e neurônios de saída (representados pelas frequências naturais). As ligações entre os neurônios das diferentes camadas são dadas por meio de pesos sinápticos  $w$ .

### 3.4.1 Modelo Sequencial

O modelo *sequencial* permite inserir camadas de uma rede neural em série, onde o *output* da primeira camada serve como *input* da segunda, e assim por diante. É possível criar um modelo sequencial passando uma lista de valores de camada diretamente entre parênteses ou adicionando camadas pelo comando “.add()”. Dentre os tipos de camadas existentes no *Keras*, a *Dense* é a mais utilizada. Ela tem como objetivo calcular uma função de ativação em conjunto com os dados de entradas e pesos.

O número de colunas selecionadas no conjunto de dados define a dimensão de cada camada. Também é necessário informar à rede o número de neurônios (variáveis de entrada) e tipo de função de ativação.

### 3.4.2 Funções de ativação

Com o advento do algoritmo de treinamento *backpropagation*, e dos métodos de atualização de pesos sinápticos por meio do gradiente, surge a necessidade de se utilizar funções de ativação com derivadas não nulas em todos os pontos. Nesse sentido, novas funções não lineares foram criadas, como a sigmoide e a *Rectified Linear Unit* (ReLU) (ROSENBLATT, 1960).

#### 3.4.2.1 Sigmoide

A função sigmoide é uma função de ativação amplamente utilizada. Na forma básica, é definida pela Equação 3, onde  $x$  é o valor de entrada do neurônio da camada intermediária ou de saída após a ponderação dos pesos sinápticos. O valor final dessa função será sempre um valor entre 0 e 1.

$$f(x) = \left[ \frac{1}{1 + e^{-ax}} \right] \quad 3$$

Essa função, cujo gráfico tem forma de S, é definida como função estritamente crescente que exibe um balanceamento adequado entre comportamento linear e não-linear (HAYKIN, 2001). Uma das grandes desvantagens dessa função, além do custo computacional, está no fato que, quando se aplica o algoritmo de *backpropagation*, os valores tendem a ficar cada vez menores. Com isso, os neurônios no início da rede neural tendem a levar mais tempo para calcular.

Por estar limitada aos valores de saída sempre positivos, entre 0 e 1, surge outro empecilho. Nem sempre se deseja que os valores enviados ao próximo neurônio sejam de sinais iguais, o que pode prejudicar o desempenho da rede.

#### 3.4.2.2 *Rectified Linear Unit – ReLU*

A função ReLU pode ser entendida como unidade linear retificada. Essa função pode ser representada pela Equação 4.

$$\text{ReLU}_{(x)} = \max\{0, x\}$$

$$\text{ReLU}_{(x)} = \begin{cases} 1, & x \geq 0 \\ 0, & \text{outro} \end{cases} \quad 4$$

Uma das maiores vantagens de se utilizar a ReLU sobre outras funções de ativação é que ela não ativa todos os neurônios ao mesmo tempo. A função converte as entradas negativas em zero e o neurônio não será ativado. Além disso, é rápido para calcular pois não há nenhuma função matemática complicada; converge mais rápido, pois a linearidade significa que a inclinação não satura para valores elevados; e não tem o problema de dissipação de gradiente devido à função identidade para valores positivos.

### 3.5 Normalização de dados

*MinMaxScaler* do *Sklearn* é uma técnica de pré-processamento que coloca os dados na mesma escala. Para cada valor em um recurso, *MinMaxScaler* subtrai o valor mínimo no recurso e, em seguida, divide pelo intervalo. O intervalo é a diferença entre o máximo original e o mínimo original. *MinMaxScaler* preserva a forma da distribuição original. Isso não altera significativamente as informações incorporadas nos dados originais. É válido ressaltar que o *MinMaxScaler* não reduz a importância de *outliers*.

O intervalo padrão para o recurso retornado por *MinMaxScaler* é de 0 a 1.

### 3.6 Compilando o modelo

Antes do modelo estar pronto para o treinamento é necessário configurar alguns parâmetros que são adicionados no processo de compilação.

A compilação serve para configurar o processo de aprendizagem. Devem ser informadas funções que definam como os pesos da rede neural são atualizados, definido pelo comando *Optimizer*. A função calcula a diferença entre os dados de teste e de validação *loss*, e uma lista de métricas a ser usada na avaliação do modelo chamada *metrics*.

O otimizador Adam é um método para otimização estocástica eficiente que requer apenas gradientes de primeira ordem com pouco requisito de memória. O método calcula taxas de aprendizagem adaptativa individuais para diferentes parâmetros de estimativas de primeiro e segundo momentos dos gradientes; o nome Adam é derivado da estimativa de momento adaptativa (KINGMA; BA, 2015).

### 3.7 *EarlyStopping*

Ao treinar redes neurais é necessário tomar decisões em relação aos hiperparâmetros utilizados para que se tenha um bom desempenho. Um desses hiperparâmetros é o número de épocas de treinamento, ou seja, quantas passagens completas do conjunto de dados (épocas) devem ser utilizadas. Se forem usadas poucas épocas, pode-se ter problemas de *underfitting* (não aprender tudo o que pode com os dados de treinamento); se usar muitas épocas, pode-se ter o problema oposto, *overfitting* (aprender demais).

Para se combater o *overfitting* é possível utilizar o *Early Stopping*, também chamado de parada antecipada. O *EarlyStopping* é um recurso de otimização utilizado para que a rede pare de treinar quando uma métrica monitorada parar de melhorar. Isso significa que, no final de cada época, deve-se calcular a precisão da classificação nos dados de validação. Quando a precisão parar de melhorar, termina-se o treinamento. Isso torna a configuração do número de épocas muito simples e automática.

### 3.8 Métrica para avaliação

Como parte do algoritmo de otimização, o erro para o estado atual do modelo deve ser estimado repetidamente. Isso requer a escolha de uma função de erro, convencionalmente chamada de função de perda, que pode ser usada para estimar a perda do modelo para que os pesos possam ser atualizados e reduzir a perda na próxima avaliação.

Os modelos de rede neural aprendem um mapeamento de entradas para saídas de exemplos e a escolha da função de perda deve corresponder ao enquadramento do problema de modelagem preditiva específico, como classificação ou regressão. Além disso, a configuração da camada de saída também deve ser apropriada para a função de perda escolhida (MACHINE LEARNING MASTERY, 2021).

#### 3.8.1 Erro médio quadrático (MSE)

Os problemas de regressão existem quando é preciso prever um valor numérico específico, o que se aplica no caso deste estudo. A perda do *Mean Square Error*, ou MSE, é a perda padrão a ser usada para problemas de regressão.

O erro quadrático médio é calculado como a média das diferenças quadradas entre os valores previstos e reais. A quadratura significa que erros maiores resultam em mais erros do

que erros menores, o que significa que o modelo é punido por cometer erros maiores (MACHINE LEARNING MASTERY, 2021).

A função de perda de erro quadrático médio pode ser usada em *Keras* especificando 'mse' ou 'mean\_squared\_error' como a função de perda ao compilar o modelo.

### 3.8.2 Erro médio absoluto (MAE)

O algoritmo utiliza as diferenças em todos os valores previstos e reais, soma-as e depois as divide pelo número de observações. Não importa se a previsão é maior ou menor do que o valor real, o algoritmo apenas olha para o valor absoluto. Um valor mais baixo indica melhor precisão.

## 3.9 Treinamento e validação da rede

Definida a arquitetura de rede e as funções de ativação a serem utilizadas, a próxima etapa do código é a de treinamento e validação.

O treinamento supervisionado é caracterizado pela presença de um “treinador” que avaliará as respostas do modelo. O treinamento consiste na apresentação de exemplos à rede para que ela aprenda os padrões de comportamento, enquanto a validação demonstra a capacidade de generalização da rede, quando forem fornecidos valores de entrada ainda não conhecidos (MENDONÇA *et al.*, 2016).

Para iniciar o treinamento da rede, é necessário um banco de dados que estejam associados à variável de resposta (saída), também conhecida como valor alvo, ou *target value*. O conjunto das variáveis de entrada e saída inseridos na rede formam um padrão, que servirão de base para o ajuste do erro. O treinamento é realizado por meio da alteração dos pesos sinápticos, em outras palavras, por meio da alteração dos valores das conexões entre os neurônios. Esse processo de ajuste só é terminado quando dentro do conjunto de treinamento, a diferença entre o valor de saída prevista pela rede e o valor alvo inserido do banco de dados é mínima.

Um dos métodos de atualização dos parâmetros internos bastante utilizado no treinamento de redes multicamada é o *backpropagation*, conforme citado anteriormente. De forma sucinta, o método funciona calculando o erro da camada de saída, comparando com o valor esperado. Os pesos entre a camada de saída e a camada anterior são atualizados para

diminuir o erro, e o gradiente do erro é repassado. Esse processo é realizado até que todos os pesos sejam atualizados.

Para validar a rede já treinada, deve-se testá-la inserindo um grupo de dados diferentes daqueles apresentados na fase de treinamento. Essa etapa é importante no processo de formulação da rede neural, uma vez que é possível verificar se a rede em treinamento está memorizando (o que é indesejável), ou está generalizando dados e mapeando relações entre entradas e saídas.

### **3.10 Predição de rede**

É possível realizar predições de valores com o modelo de regressão finalizado, chamando a função *Predict()*. Essa função obtém uma matriz de uma ou mais instâncias de dados, os quais podem ser comparados com os valores do banco de dados.

## 4 RESULTADOS

### 4.1 Arquitetura de rede

As rotinas implementadas em linguagem *Python*, tanto para as lajes nervuradas quanto para as lajes pré-moldadas estão disponíveis no Apêndice A.

#### 4.1.1 Lajes nervuradas

Para essa rede, a camada de entrada é composta por 11 neurônios, conforme segue:

- a) Neurônio 1: Condição de contorno;
- b) Neurônio 2: Comprimento da laje (L);
- c) Neurônio 3: Largura da laje (B);
- d) Neurônio 4: Massa (kg);
- e) Neurônio 5: Resistência do concreto ( $f_{ck}$ );
- f) Neurônio 6: Espaçamento entre nervuras (S);
- g) Neurônio 7: Largura da nervura (b);
- h) Neurônio 8: Espessura da camada concreto ( $h_c$ );
- i) Neurônio 9: Altura total da laje (h);
- j) Neurônio 10: Altura da nervura ( $h_t$ );
- k) Neurônio 11: Espessura da nervura ( $h_{eq}$ );

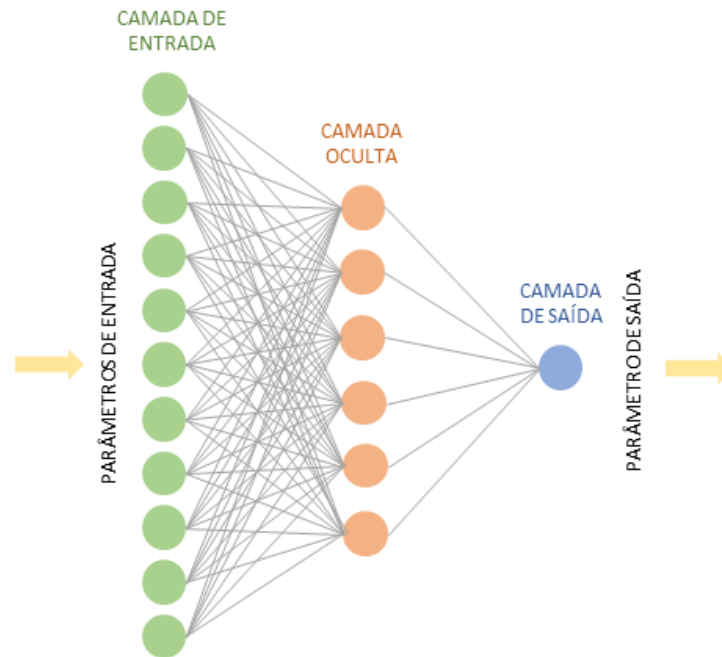
Tomando como base as recomendações para a determinação do número de neurônios citadas anteriormente, se pode estimar um número dos demais parâmetros de arquitetura, sendo:

- a) Número de neurônios na camada de entrada: 11;
- b) Número de neurônios nas camadas ocultas: entre 6 e 12;
- c) Número de neurônios na camada de saída: 1.

Para a camada de saída será atribuído um único neurônio, representando a frequência natural da laje. O esquema da rede neural pode ser observado na Figura 6.



Figura 6 - Arquitetura de lajes nervuradas proposta da RNA



Fonte: O autor (2022)

#### 4.1.2 Lajes pré-moldadas

Para essa rede, a camada de entrada é composta por 9 neurônios, conforme segue:

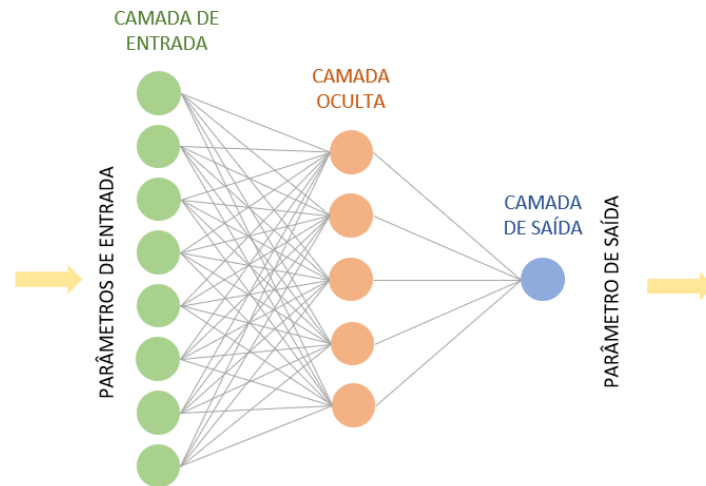
- a) Neurônio 1: Condição de contorno;
- b) Neurônio 2: Comprimento da laje (L);
- c) Neurônio 3: Largura da laje (B);
- d) Neurônio 4: Resistência do concreto (fck);
- e) Neurônio 5: Espaçamento entre vigotas (S);
- f) Neurônio 6: Espessura da camada concreto (hc);
- g) Neurônio 7: Altura total da laje (h);
- h) Neurônio 8: Altura da vigota (ht);
- i) Neurônio 9: Espessura da vigota (heq);

Tomando como base as recomendações para a determinação do número de neurônios citadas anteriormente, se pode estimar um número dos demais parâmetros de arquitetura, sendo:

- a) Número de neurônios na camada de entrada: 9;
- b) Número de neurônios nas camadas ocultas: entre 5 e 10;
- c) Número de neurônios na camada de saída: 1.

Para a camada de saída será atribuído um único neurônio, representando a frequência natural da laje. O esquema da rede neural pode ser observado na Figura 7.

Figura 7 - Arquitetura de lajes pré-moldadas proposta da RNA



Fonte: O autor (2022)

## 4.2 Banco de dados

O arquivo do banco de dados foi formatado em arquivo CSV (*comma-separated values*) para que o software pudesse interpretar. A representação está inserida na Figura 8.

Figura 8 - Dados csv

1,633,303,330,30,65,7,5,26,21,16.24,28.52	
2,889,488,330,30,65,7,5,26,21,16.24,25.78	
3,1091,363,330,30,65,7,5,26,21,16.24,32.29	
3,702,492,330,30,65,7,5,26,21,16.24,31.64	
2,672,370,330,30,65,7,5,26,21,16.24,39.19	
3,861,303,330,30,65,7,5,26,21,16.24,31.7	
4,1005,488,330,30,65,7,5,26,21,16.24,24.22	
1,1005,383,330,30,65,7,5,26,21,16.24,29.95	
3,421,353,330,30,65,7,5,26,21,16.24,44.53	
2,640,419,330,30,65,7,5,26,21,16.24,32.42	
3,408,363,250,25,59,9,5,17,12,11.57,31.25	
3,408,318,250,25,59,9,5,17,12,11.57,41.8	
2,339,188,250,25,59,9,5,17,12,11.57,71.35	
5,353,338,250,25,59,9,5,17,12,11.57,38.15	
2,473,353,250,25,59,9,5,17,12,11.57,47.4	
1,338,210,250,25,59,9,5,17,12,11.57,56.77	
1,218,128,250,25,59,9,5,17,12,11.57,117.02	
3,408,318,250,25,59,9,5,17,12,11.57,49.48	
5,353,338,250,25,59,9,5,17,12,11.57,43.23	
3,353,202,250,25,59,9,5,17,12,11.57,47.4	
1,338,218,250,25,59,9,5,17,12,11.57,67.71	
1,1252,805,285,25,60,10,5,20,15,13.9,7.82	
3,965,805,285,25,60,10,5,20,15,13.9,7.82	
1,960,805,285,25,60,10,5,20,15,13.9,7.04	
1,965,805,285,25,60,10,5,20,15,13.9,7.84	
3,960,805,285,25,60,10,5,20,15,13.9,7.93	
	1,517,331,20,42,4,12,8,9.43,36.92
	1,266,246,20,42,4,12,8,9.43,47.14,
	1,317,295,20,42,4,12,8,9.43,59.38,
	1,310,185,20,42,4,12,8,9.43,36.46,
	4,326.5,256,20,42,4,12,8,9.43,36.46,
	4,404,326.5,20,42,4,12,8,9.43,30.15,
	1,358.5,336,20,42,4,12,8,9.43,65.1,
	1,696,527.5,25,49,4,20,16,15.1,16.62,
	4,716,665,25,49,4,20,16,15.1,14.58,
	1,696,527.5,25,49,4,20,16,15.1,15.36,
	4,866,432.5,25,49,4,20,16,15.1,21.61,
	1,671,562.5,25,49,4,20,16,15.1,17.97,
	1,671,562.5,25,49,4,20,16,15.1,17.06,
	1,921,452.5,25,49,4,20,16,15.1,14.84,
	4,866,867.5,25,49,5,25,20,18.87,14.64,
	1,921,452.5,25,49,4,20,16,15.1,23.05,
	1,889,457,25,49,4,20,16,15.1,33.75,
	4,389.5,386,25,49,4,20,16,15.1,25.52,
	4,482.5,519.5,25,49,4,20,16,15.1,34.77,
	4,540,333,25,49,4,20,16,15.1,35.94,
	4,543.5,540,25,49,4,20,16,15.1,31.51,
	4,500.5,305.5,25,49,4,20,16,15.1,57.94,
	1,340,196,25,49,5,21,16,16.45,60.94,
	6,434,389,25,49,5,13,8,10.22,28.26,
	6,200,191,25,49,5,13,8,10.22,95.24,
	6,230,121,25,49,5,13,8,10.22,62.62,]

Fonte: O autor (2022)

### 4.3 Normalização de dados

A Figura 9 e a Figura 10 mostram o resultado da normalização de dados realizada pela rede, pelo método do *minmaxscaler*. É possível verificar que se obtiveram valores entre 0 e 1 conforme esperado para ambos os tipos de laje. Nessas colunas, cada coluna corresponde ao valor de um parâmetro de entrada, seguindo a sequência adotada.

Figura 9 - Dados normalizados para lajes nervuradas pelo *minmaxscaler*

MinMaxScaler()											
[0.	0.40135397	0.25849335	1.	1.	1.	0.	0.	1.	1.	1.	0.19530824]
[0.25	0.64893617	0.53175775	1.	1.	1.	0.	0.	1.	1.	1.	0.17039462]
[0.5	0.844294	0.34711965	1.	1.	1.	0.	0.	1.	1.	1.	0.2295872 ]
[0.5	0.46808511	0.53766617	1.	1.	1.	0.	0.	1.	1.	1.	0.22367703]
[0.25	0.43907157	0.35745938	1.	1.	1.	0.	0.	1.	1.	1.	0.29232588]
[0.5	0.62185687	0.25849335	1.	1.	1.	0.	0.	1.	1.	1.	0.22422259]
[0.75	0.76112186	0.53175775	1.	1.	1.	0.	0.	1.	1.	1.	0.15621022]
[0.	0.76112186	0.37666174	1.	1.	1.	0.	0.	1.	1.	1.	0.2083106 ]
[0.5	0.19632495	0.3323486	1.	1.	1.	0.	0.	1.	1.	1.	0.34088016]
[0.25	0.40812379	0.42983752	1.	1.	1.	0.	0.	1.	1.	1.	0.23076923]
[0.5	0.18375242	0.34711965	0.	0.	0.	0.66666667	0.	0.	0.	0.	0.22013093]
[0.5	0.18375242	0.28064993	0.	0.	0.	0.66666667	0.	0.	0.	0.	0.31605746]
[0.25	0.11702128	0.08862629	0.	0.	0.	0.66666667	0.	0.	0.	0.	0.58474268]
[1.	0.13056093	0.31019202	0.	0.	0.	0.66666667	0.	0.	0.	0.	0.28286961]
[0.25	0.24661509	0.3323486	0.	0.	0.	0.66666667	0.	0.	0.	0.	0.36697581]
[0.	0.11605416	0.1211226	0.	0.	0.	0.66666667	0.	0.	0.	0.	0.45217312]
[0.	0.	0.	0.	0.	0.	0.66666667	0.	0.	0.	0.	1. ]
[0.5	0.18375242	0.28064993	0.	0.	0.	0.66666667	0.	0.	0.	0.	0.38588834]
[1.	0.13056093	0.31019202	0.	0.	0.	0.66666667	0.	0.	0.	0.	0.32905983]
[0.5	0.13056093	0.10930576	0.	0.	0.	0.66666667	0.	0.	0.	0.	0.36697581]
[0.	0.11605416	0.13293944	0.	0.	0.	0.66666667	0.	0.	0.	0.	0.55164575]
[0.	1.	1.	0.4375	0.	0.16666667	1.	0.	0.33333333	0.33333333	0.49892934	0.0070922 ]
[0.5	0.72243714	1.	0.4375	0.	0.16666667	1.	0.	0.33333333	0.33333333	0.49892934	0.0070922 ]
[0.	0.71760155	1.	0.4375	0.	0.16666667	1.	0.	0.33333333	0.33333333	0.49892934	0. ]
[0.	0.72243714	1.	0.4375	0.	0.16666667	1.	0.	0.33333333	0.33333333	0.49892934	0.00727405]
[0.5	0.71760155	1.	0.4375	0.	0.16666667	1.	0.	0.33333333	0.33333333	0.49892934	0.00809238]]

Fonte: O autor (2022)

Figura 10 - Dados normalizados para lajes pré-moldadas pelo *minmaxscaler*

```

MinMaxScaler()
[[0 0.439667 0.281313 0 0 0 0 0 0 0.276965]
 [0 0.0915395 0.167448 0 0 0 0 0 0 0.40367]
 [0 0.162275 0.233088 0 0 0 0 0 0 0.555418]
 [0 0.152566 0.0857334 0 0 0 0 0 0 0.271262]
 [0.6 0.175451 0.180844 0 0 0 0 0 0 0.271262]
 [0.6 0.28294 0.275285 0 0 0 0 0 0 0.193032]
 [0 0.219834 0.288011 0 0 0 0 0 0 0.626333]
 [0 0.687933 0.544541 1 1 0 0.615385 0.666667 0.600636 0.0252913]
 [0.6 0.715673 0.728734 1 1 0 0.615385 0.666667 0.600636 0]
 [0 0.687933 0.544541 1 1 0 0.615385 0.666667 0.600636 0.00967022]
 [0.6 0.923717 0.417281 1 1 0 0.615385 0.666667 0.600636 0.087156]
 [0 0.653259 0.591427 1 1 0 0.615385 0.666667 0.600636 0.0420283]
 [0 0.653259 0.591427 1 1 0 0.615385 0.666667 0.600636 0.0307463]
 [0 1 0.444072 1 1 0 0.615385 0.666667 0.600636 0.00322341]
 [0.6 0.923717 1 1 1 1 1 1 1 0.000743863]
 [0 1 0.444072 1 1 0 0.615385 0.666667 0.600636 0.105009]
 [0 0.955617 0.4501 1 1 0 0.615385 0.666667 0.600636 0.237664]
 [0.6 0.262829 0.35499 1 1 0 0.615385 0.666667 0.600636 0.135631]
 [0.6 0.391817 0.533825 1 1 0 0.615385 0.666667 0.600636 0.25031]
 [0.6 0.471567 0.283992 1 1 0 0.615385 0.666667 0.600636 0.264815]
 [0.6 0.476422 0.561286 1 1 0 0.615385 0.666667 0.600636 0.209893]
 [0.6 0.416782 0.247153 1 1 0 0.615385 0.666667 0.600636 0.537565]
 [0 0.194175 0.100469 1 1 1 0.692308 0.666667 0.743644 0.574758]
 [1 0.324549 0.359009 1 1 1 0.0769231 0 0.0836864 0.169601]
 [1 0 0.0937709 1 1 1 0.0769231 0 0.0836864 1]
 [1 0.0416089 0 1 1 1 0.0769231 0 0.0836864 0.595586]]

```

Fonte: O autor (2022)

#### 4.4 Definindo parâmetros da arquitetura na rede

Após a normalização dos dados, é necessário informar à rede quais os parâmetros de arquitetura de rede adotados até então. Por meio do comando *trainX* e *trainY* delimita-se as variáveis de entrada (X) e saída (Y) na etapa de treinamento, relacionado à respectiva coluna em que se encontra.

Para separar o conjunto de teste e validação foi arbitrado o percentual de 30%, onde 70% dos valores de treinamento são armazenados para a fase de teste e o restante para a validação da rede.

Tanto para a rede das lajes nervuradas quanto para as pré-moldadas, a arquitetura das redes no que diz respeito à quantidade variáveis de entrada e saída seguiu o esquema apresentado na Figura 6 e Figura 7.

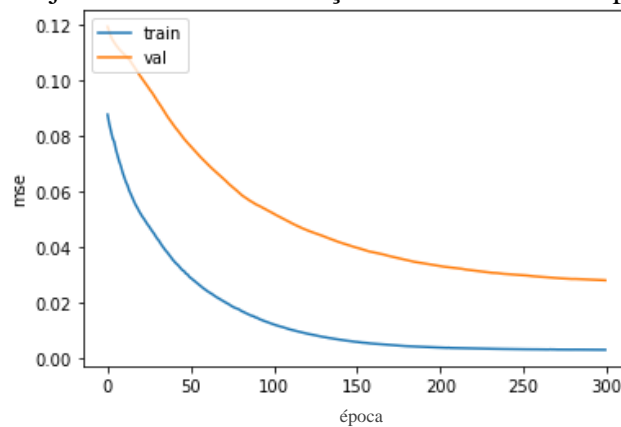
A etapa de treinamento finaliza com a aplicação do *earlystopping*.

## 4.5 Métricas de avaliação

### 4.5.1 Erro médio quadrático (MSE) e Erro médio absoluto (MAE)

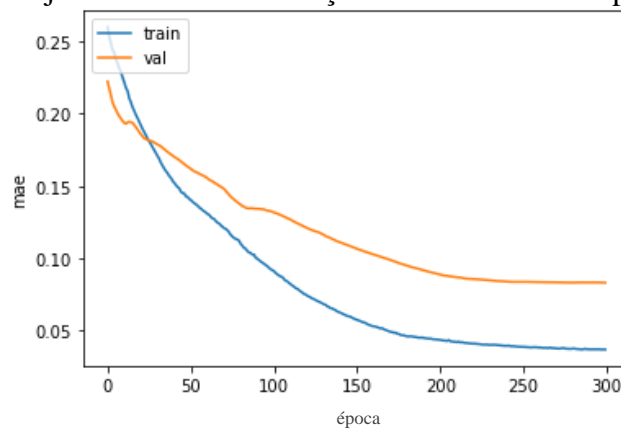
Como exemplo, os resultados de treinamento e validação de uma das redes criadas podem ser observados na Figura 11 e na Figura 12.

Figura 11 – Lajes nervuradas: Evolução do MSE em 300 épocas



Fonte: O autor (2022)

Figura 12 - Lajes nervuradas: Evolução do MAE em 300 épocas

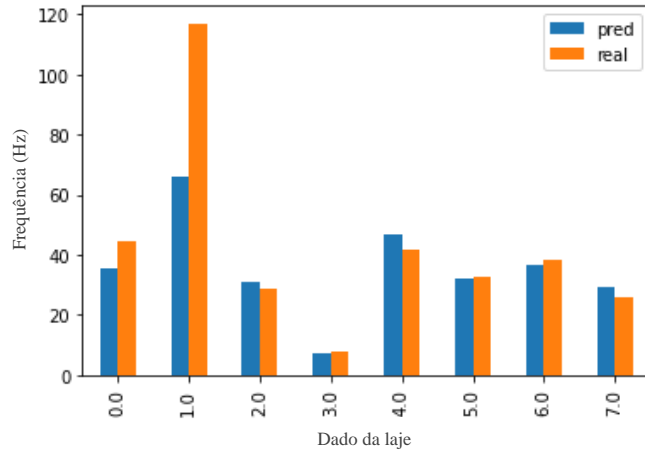


Fonte: O autor (2022)

É possível verificar para ambas as métricas de avaliação que o erro, apresentado tanto na fase de treinamento quanto na de validação, tende a ser reduzido com o aumento no número de épocas. Vale salientar que em determinado momento da curvatura é apresentado uma estabilização do erro, entre as épocas de 250 e 300, para ambos os casos. Para a fase de teste dessa rede, apresentando a ela dados que não foram vistos durante as etapas de treinamento e validação, é possível verificar na Figura 13 que para resultados com maiores frequências a rede

não foi capaz de prever de forma eficaz. Analisando o banco de dados, é possível verificar que existem poucas lajes de frequências elevadas, sendo essa a possível causa para a dificuldade da rede em apresentar uma predição adequada nesses níveis de frequência.

Figura 13 - Lajes nervuradas: Predição da rede na fase de teste com 300 épocas



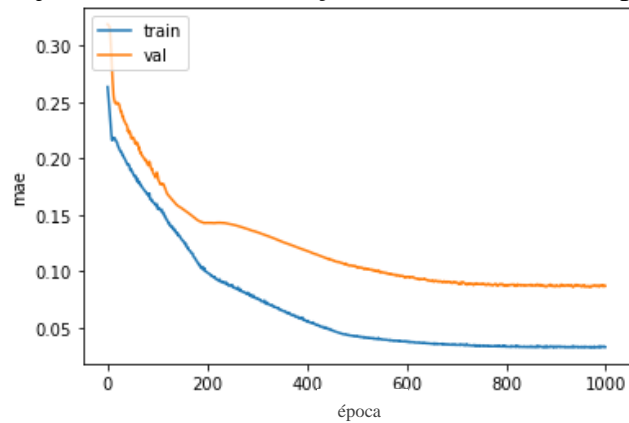
Fonte: O autor (2022)

Mesmo o erro mostrando uma certa estabilidade da curvatura em épocas mais avançadas, ainda não é possível assegurar o comportamento do erro mínimo. Sendo assim, optou-se por avaliar a rede novamente utilizando 1000 épocas.

Essa decisão ocasiona aumento no tempo despendido para a finalização das etapas, porém é importante para que se consiga o ponto ótimo no que diz respeito ao tempo de operação e complexidade de cálculo da rede.

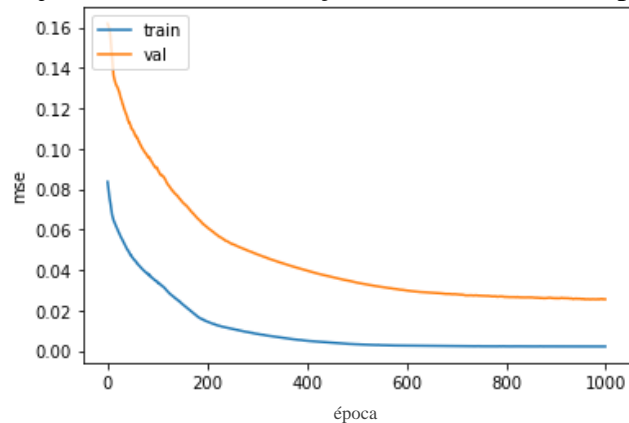
Avaliando a Figura 14 e a Figura 15, pode-se verificar que a rede apresenta estabilidade do erro por volta das 800 épocas. Ainda assim, o *earlystopping* não foi ativado, uma vez que a rede só finalizou o treinamento ao final das 1000 épocas. Isso significa que se pode obter uma estabilidade do erro ainda maior com um maior número de épocas arbitrado.

Figura 14 - Lajes nervuradas: Evolução do MAE em 1000 épocas



Fonte: O autor (2022)

Figura 15 - Lajes nervuradas: Evolução do MSE em 1000 épocas

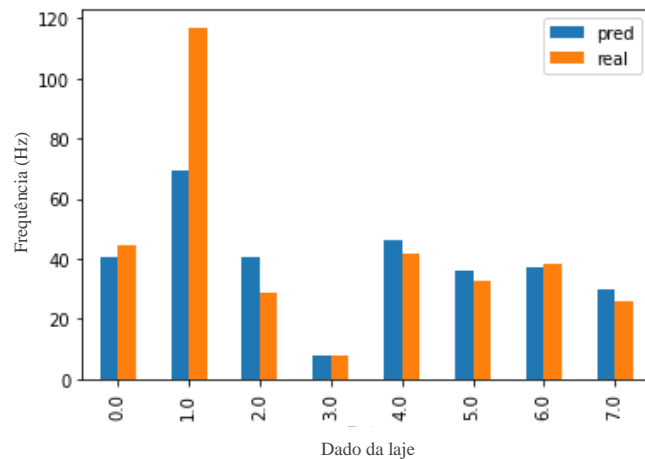


Fonte: O autor (2022)

Como a determinação inicial do número de épocas é feita de forma arbitrária pelo operador, e o treinamento é supervisionado, esses procedimentos são importantes e servem como base para a tomada de decisão mais precisa.

A predição da rede com 1000 épocas é evidenciada na Figura 16. Pode-se perceber que em comparação à rede treinada com 300 épocas, a capacidade de previsão foi maior nessa rede. Os dados das lajes com as menores frequências possuem uma capacidade de aproximação maior entre o previsto e o real. Já para as lajes com frequências maiores a aproximação do previsto com o real também foi maior.

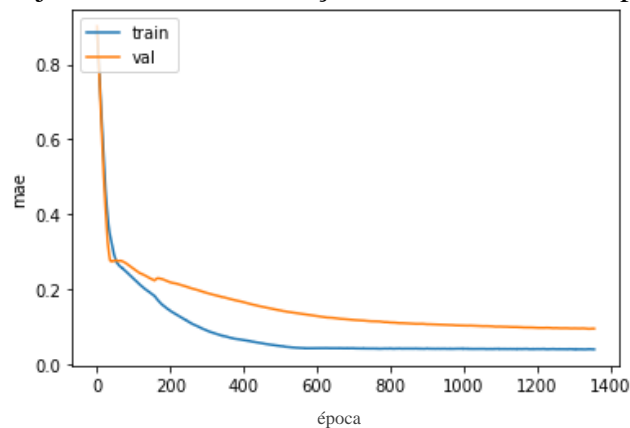
Figura 16 - Lajes nervuradas: Predição da rede na fase de teste com 1000 épocas



Fonte: O autor (2022)

Em virtude de o *earlystopping* ainda não ter sido ativado, se estipulou um aumento no número de épocas para 3000. Com esse aumento foi possível notar que a rede finalizou o processo de treinamento automaticamente na época 1355, com um erro na ordem de 0,002. É nesse momento que a rede restaura os pesos sinápticos do modelo a partir do final da melhor época e salva esses dados para que possam ser utilizados posteriormente. É possível acompanhar os gráficos de erro na Figura 17 e Figura 18.

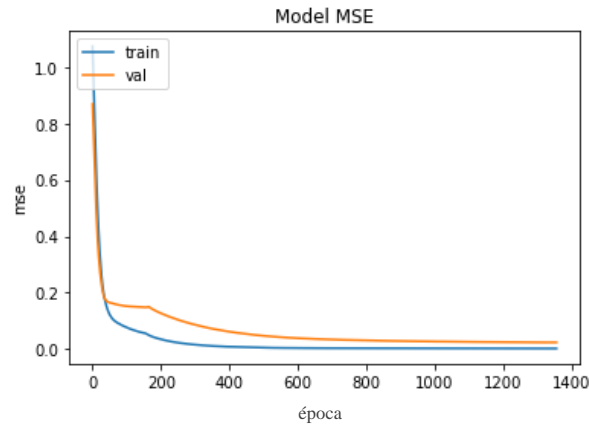
Figura 17 - Lajes nervuradas: Evolução do MAE em 3000 épocas



Fonte: O autor (2022)



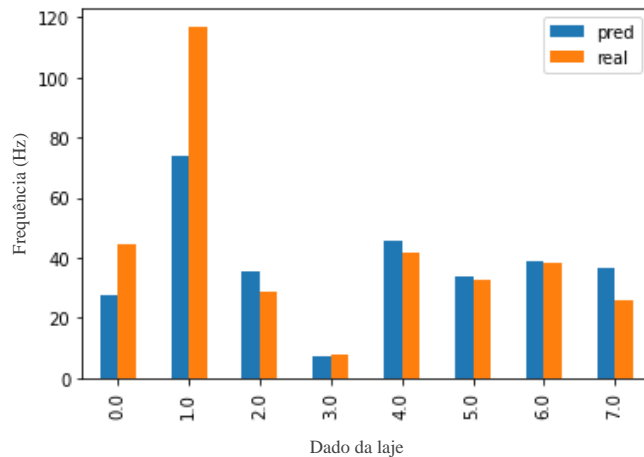
Figura 18 - Lajes nervuradas: Evolução do MSE em 3000 épocas



Fonte: O autor (2022)

A Figura 19 mostra a predição da rede com 3000 épocas, quando comparada aos valores reais já conhecidos do banco de dados. É possível verificar um comportamento semelhante de capacidade de previsão à rede com 1000 épocas.

Figura 19 - Lajes nervuradas: Predição da rede na fase de teste com 3000 épocas



Fonte: O autor (2022)

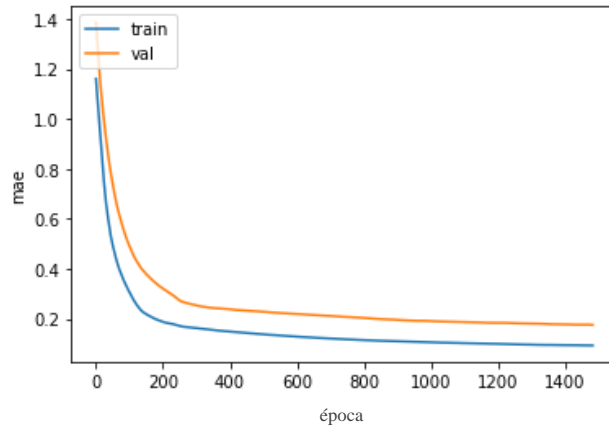
Como as variáveis de entrada alteram de acordo com o tipo de laje, uma vez que lajes nervuradas apresentam características geométricas específicas das nervuras que as pré-moldadas não têm, é importante alterar alguns parâmetros da arquitetura da rede para se obter o melhor resultado.

A estrutura geral da rede foi mantida a mesma da utilizada para as lajes nervuradas, alterando somente o número de neurônios na camada de entrada e na camada oculta.

O número de épocas para o treinamento dessa rede foi estipulado em 3000 logo no início em função de já se esperar que a rede apresente uma ativação do *earlystopping* por volta desse

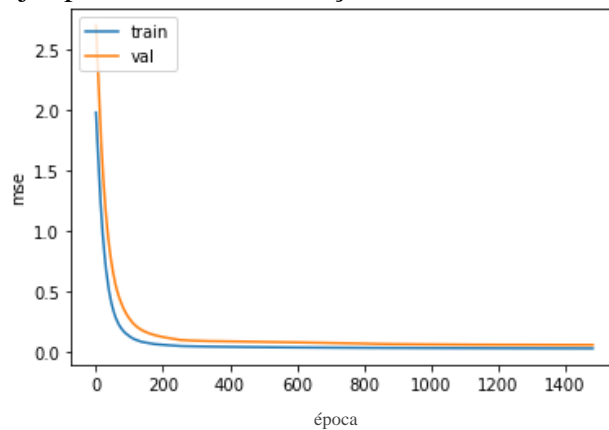
valor. A Figura 20 e a Figura 21 mostram o comportamento do MAE e do MSE para o valor de 3000 épocas.

Figura 20 - Lajes pré-moldadas: Evolução do MAE em 3000 épocas



Fonte: O autor (2022)

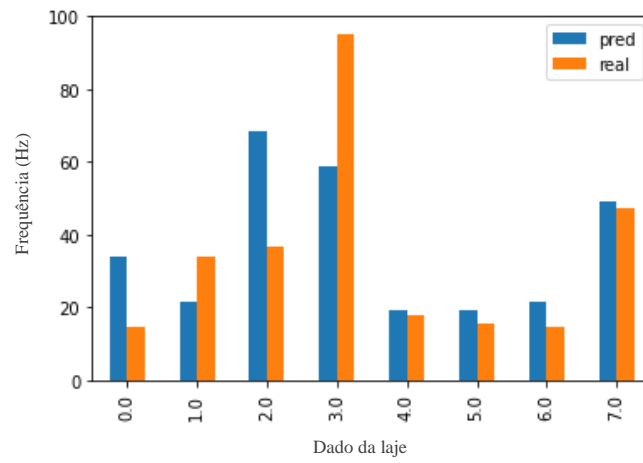
Figura 21 - Lajes pré-moldadas: Evolução do MSE em 3000 épocas



Fonte: O autor (2022)

É interessante notar que dessa vez o *earlystopping* foi atingido na época 1484, finalizando o treinamento. O valor de erro não aprimorou de 0.05618, e o valor dos pesos sinápticos para essa rede treinada foi automaticamente salvo, por meio do código de programação. A Figura 22 apresenta um comparativo da predição da rede com os valores reais separados para a etapa de validação. Pode-se notar que o mesmo foi observado nas lajes nervuradas, onde frequências mais baixas, por possuírem mais dados no banco de dados com leituras *in loco*, conseguiram ser previstas de forma mais aproximada.

Figura 22 - Lajes pré-moldadas: Predição da rede na fase de teste com 3000 épocas



Fonte: O autor (2022)

## 5 CONCLUSÕES

Esta pesquisa apresentou um estudo inicial sobre o uso de redes neurais artificiais para a previsão da frequência natural em lajes de concreto armado. Os resultados indicam que as redes foram capazes de realizar a predição das frequências naturais, salvo algumas considerações.

A aplicação das redes nesta pesquisa foi realizada em fases detalhadas, de forma a servir como base para futuros experimentos. As redes contaram com o levantamento das informações, preparação do banco de dados e definição dos parâmetros de entrada.

Para a fase de treinamento, validação e teste foram utilizados 11 fatores a serem empregados como variáveis de entrada, sendo eles: condição de contorno, comprimento da laje, largura da laje, massa, resistência do concreto, espaçamento entre as nervuras, largura da nervura, espessura da camada de concreto, altura final da laje, altura da nervura e espessura da nervura, para as lajes nervuradas, e 9 fatores para as lajes pré-moldadas: condições de contorno, comprimento da laje, largura da laje, resistência do concreto, espaçamento entre vigotas, espessura da camada de concreto, altura final da laje, altura da vigota, espessura da vigota.

Foram utilizados 70% dos dados disponíveis para a etapa de treinamento e 30% para validação. Nessa fase, observou-se que somente com o aumento de 300 para 3000 épocas a utilização da ferramenta de *earlystopping* foi eficaz em determinar o número ideal de épocas. Como o processo de determinação do número de épocas é de forma arbitrária, a metodologia de comparação de resultados pôde guiar a rede a obter maior precisão.

Como conclusão, é possível afirmar que a rede possibilita a previsão de frequências de forma satisfatória para as lajes de baixas frequências. Para as lajes com frequências mais elevadas a rede não foi capaz de prever de forma precisa, uma vez que se detectou uma necessidade de manipulação de dados faltantes no banco de dados. Vale salientar que o problema de vibrações excessivas em lajes ocorre em frequências mais baixas, as quais estão mais próximas da frequência de excitação pelas atividades humanas.

Para a elaboração de futuros estudos de aperfeiçoamento da presente pesquisa, sugere-se: a avaliação dos hiperparâmetros utilizados na pesquisa e a inserção de novos hiperparâmetros que possam ser otimizados para melhorar a predição da rede; determinar a influência das variáveis de entrada na variável de saída, para que dessa forma seja possível excluir a necessidade de se ter todas as variáveis; vincular esses e demais tipos de lajes disponíveis no mercado em uma única plataforma; disponibilizar a rede como uma ferramenta

alternativa aos profissionais, de forma fácil e rápida na previsão de frequências naturais, utilizando de poucas variáveis de entrada e baixos custos com equipamentos.

## REFERÊNCIAS

- ABAMBRES, Miguel; LANTSOGHT, Eva O.L. Neural network-based formula for shear capacity prediction of one-way slabs under concentrated loads. **Engineering Structures**, [s. l.], v. 211, p. 110501, 2020. Disponível em: <https://doi.org/10.1016/j.engstruct.2020.110501>
- AHMED, Mezgeen Sisen. **Damage Detection in Reinforced Concrete Square Slabs Using Modal Analysis and Artificial Neural Network**. [s. l.], n. March, 2016.
- AMIT BANERJEE, G POHIT, Brajesh Panigrahi. Vibration Analysis and Prediction Natural Frequencies of Cracked.pdf. **Science Direct**, [s. l.], 2016.
- ANACONDA. **Anaconda**. [S. l.], 2021. Disponível em: <https://www.anaconda.com/>.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 14860-1 - Laje pré-fabricada - pré-laje - requisitos**.Rio de Janeiro: [s. n.], 2002.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6118 - Projeto de estruturas de concreto**.Rio de Janeiro: [s. n.], 2014.
- BLEVINS, Robert D. **Formulas for Dynamics, Acoustics and Vibration**. 1. ed. [S. l.: s. n.], 2016.
- BRITO JÚNIOR, C.A.R. *et al.* Redes neurais artificiais aplicadas para a predição do comportamento dinâmico-mecânico de compósitos de matriz epóxi reforçados com fibras de carbono. **Matéria (Rio de Janeiro)**, [s. l.], v. 12, n. 2, p. 346–357, 2007. Disponível em: <https://doi.org/10.1590/s1517-70762007000200013>
- CAVALCANTI, Rosely de Souza. **Ensaio De Ondas Ultrassônicas E Redes Neurais Artificiais Na Avaliação da Resistência à Compressão do Concreto**. [s. l.], p. 179, 2010.
- COSTA, João Henrique Rodrigues. **PREVISÃO DE VIBRAÇÕES USANDO REDES NEURAS ARTIFICIAIS NO CONTROLE DE DESMONTE DE ROCHAS**. [s. l.], 2019.
- DEEP LEARNING BOOK. **O neurônio biológico e matemático**. [S. l.], [s. d.]. Disponível em: <https://www.deeplearningbook.com.br/o-neuronio-biologico-e-matematico/>. Acesso em: 15 dez. 2021.
- FONSECA, Elaine Toscano. **Comportamento de Vigas de Aço Sujeitas a Cargas Concentradas Através de Técnicas de Inteligência Computacional**. [s. l.], 2003.
- FORUM, World Economic. **The concrete that heals its own cracks**. [S. l.], 2016. Disponível em: <https://www.weforum.org/agenda/2016/01/the-concrete-that-heals-its-own-cracks/>.
- GCCA. **Concrete and Sustainability**. [S. l.], 2019. Disponível em: <https://gccassociation.org/concrete-and-sustainability/>. Acesso em: 29 set. 2020.
- GOMES, F.P. Curso de estatística experimental. **FEALQ**, [s. l.], 2009.

- GONZÁLEZ DUQUE, Raúl. Python para todos. **Web book**, [s. l.], n. 6, p. 108, 2000.  
Disponível em: <http://dspace.universia.net/handle/2024/919>
- HAJEK, Petr. Contribution of concrete structures to sustainability - Challenge for the future. **IOP Conference Series: Materials Science and Engineering**, [s. l.], v. 442, n. 1, p. 0–10, 2019. Disponível em: <https://doi.org/10.1088/1757-899X/442/1/012013>
- HAYKIN, Simon. **Redes Neurais - Princípios e Prática [2ª Ed][Haykin][2001].pdf**. [S. l.: s. n.], 2001.
- HERBERT M. GOMES. Redes Neurais Artificiais Para Detecção De Danos. **Mecanica Computacional Vol XVIII**, [s. l.], v. XXIII, n. November, p. 1–17, 2004.
- KERAS. [S. l.], 2021. Disponível em: <https://keras.io/>.
- KINGMA, Diederik P.; BA, Jimmy Lei. Adam: A method for stochastic optimization. **3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings**, [s. l.], p. 1–15, 2015.
- KONDRU, Lokesh; RAO, Narasinga. Damage Detection in Cantilever Beams Using Artificial Neural Networks. **International Journal of Mechanical and Production Engineering Research and Development (IJMPERD)**, [s. l.], v. 3–1, n. 1, p. 259–268, 2013.
- LAGO, Fernanda Scussiato. **Estudo numérico e experimental de parâmetros de vibração em lajes de concreto**. 2019. - Universidade de Passo Fundo, [s. l.], 2019.
- LEHNE, Johanna; PRESTON, Felix. **Making Concrete Change Innovation in Low-carbon Cement and Concrete The Royal Institute of International Affairs**, Chatham House Report Series, [www.chathamhouse.org/sites/default/files/publications/research/2018-06-13-makingconcrete-c](http://www.chathamhouse.org/sites/default/files/publications/research/2018-06-13-makingconcrete-c). [s. l.], p. 138, 2018. Disponível em: [www.chathamhouse.org](http://www.chathamhouse.org)
- MACHINE LEARNING MASTERY. [S. l.], 2021. Disponível em: <https://machinelearningmastery.com/>.
- MEHRJOO, M. *et al.* Damage detection of truss bridge joints using Artificial Neural Networks. **Expert Systems with Applications**, [s. l.], v. 35, n. 3, p. 1122–1131, 2008. Disponível em: <https://doi.org/10.1016/j.eswa.2007.08.008>
- MENDONÇA, Jorge Artur França de *et al.* **APLICAÇÃO DE REDES NEURAIAS ARTIFICIAIS NA OTIMIZAÇÃO DE PLACAS LAMINADAS**. [s. l.], p. 2330–2341, 2016.
- MILLER, Sabbie A.; HORVATH, Arpad; MONTEIRO, Paulo J.M. Impacts of booming concrete production on water resources worldwide. **Nature Sustainability**, [s. l.], v. 1, n. 1, p. 69–76, 2018. Disponível em: <https://doi.org/10.1038/s41893-017-0009-5>
- MOTA, David A. Fanella; Mike. Vibration of Reinforced Concrete Floor Systems. **Journal of Chemical Information and Modeling**, [s. l.], v. 53, n. 9, p. 1689–1699, 2013. Disponível em: <https://doi.org/10.1017/CBO9781107415324.004>
- NAGALLI, A. Comparativo tecno-econômico entre obras comerciais com características

- sustentáveis e convencionais. **Téchne**, [s. l.], p. 60–63, 2012.
- OLIVEIRA, Fernando Artur Nogueira Silva; Maria Eduarda Maia Ferreira Gomes; Romilde Almeida de. **Dimensionamento de pilares de concreto armado de seções retangular e circular maciça submetidos à flexão oblíqua composta utilizando redes neurais artificiais**. [s. l.], v. 25, n. 27, p. 27–33, 2016.
- PARKER, D. B. Learning Logic, Invention Report. **Office of Technology Licensing**, [s. l.], 1982.
- PINHEIRO, LIBANIO M., MUZARDO, CASSIANE D., SANTOS, Sandro P.
- FUNDAMENTOS DO CONCRETO E PROJETO DE EDIFÍCIOS. *In*: [S. l.: s. n.], 2010.
- RAO, Singiresu. **Vibrações mecânicas. Tradução de Arlete Simille Marques**. [S. l.: s. n.], 2008.
- ROSENBLATT, Frank. Perceptron Simulation Experiments. **Proceedings of the IRE**, [s. l.], v. 48, n. 3, p. 301–309, 1960. Disponível em: <https://doi.org/10.1109/JRPROC.1960.287598>
- SAHIN, M.; SHENOI, R. A. Quantification and localisation of damage in beam-like structures by using artificial neural networks with experimental validation. **Engineering Structures**, [s. l.], v. 25, n. 14, p. 1785–1802, 2003. Disponível em: <https://doi.org/10.1016/j.engstruct.2003.08.001>
- SILVA, Clarence W. de. **Vibration - Fundamentals and Practice**. 2. ed. [S. l.]: CRC Press, 2006.
- SILVA, J. G. S.; GASPAR, C.M.R.; LOPES, E. D. C. Análise Dinâmica e Controle de Vibrações de Pisos Mistos (Aço-Concreto) Submetidos a Atividades Humanas Rítmicas. **Metálica: Revista da Associação Portuguesa de Construção Metálica e Mista**, [s. l.], v. 34, p. 6–12, 2014.
- SINDICATO NACIONAL DA INDÚSTRIA DO CIMENTO. **SNIC**. [S. l.], 2020. Disponível em: <http://snic.org.br>. Acesso em: 29 set. 2020.
- SORIANO, Humberto L. **Introdução à Dinâmica das Estruturas**. 1. ed. Rio de Janeiro: [s. n.], 2014.
- SPYDER. **Spyder**. [S. l.], 2021. Disponível em: <https://www.spyder-ide.org/>.
- WATTS, Jonathan. **Concrete: the most destructive material on Earth**. [S. l.], 2019. Disponível em: <https://www.theguardian.com/cities/2019/feb/25/concrete-the-most-destructive-material-on-earth>. Acesso em: 29 set. 2020.
- ZUBEN, Fernando José Von. **Modelos paramétricos e não-paramétricos de redes neurais artificiais e aplicações**. [S. l.: s. n.], 1996.



## APÊNDICE A – ALGORITMO DA REDE NEURAL ARTIFICIAL (RNA)

### RNA LAJES NERVURADAS

```

from tensorflow import keras
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from numpy import loadtxt
from keras.models import Sequential
from keras.layers import Dense
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn import metrics
from keras.callbacks import EarlyStopping, ModelCheckpoint, CSVLogger

#PREPARANDO OS DADOS
dataset_real = loadtxt('bancox.csv', delimiter=',')
    # Usa essa função para normalizar de 0 a 1
scaler = MinMaxScaler(feature_range=(0, 1))
dataset = scaler.fit_transform(dataset_real)

#DIVIDINDO ENTRE VARIÁVEIS DE ENTRADA (X) E DE SAÍDA (Y)
trainX = dataset[:,0:11] #PRIMEIRAS COLUNAS, DE 0 A N (EX.: 0:8 PARA 7 COLUNAS)
trainY = dataset[:,11] #ÍNDICE QUE SE ENCONTRA A VARIÁVEL DE SAÍDA (EX.: ÍNDICE 8, COLUNA
9)

#SEPARANDO CONJUNTO TESTE E VALIDAÇÃO
X_train, X_test, y_train, y_test = train_test_split(
    trainX, trainY, test_size=0.3, random_state=42) # test_size: PERCENTUAL USADO PARA TESTE

#DEFININDO O MODELO KERAS
#Número de camadas
model = Sequential()
model.add(Dense(6, input_dim=11, activation='relu')) #INPUT_DIM: VARIÁVEIS DE ENTRADA, DENSE: N
CAMADAS INTERMEDIARIAS
model.add(Dense(1, activation='relu')) #CAMADA DE SAÍDA

#EARLYSTOPPING
monitor = EarlyStopping(monitor = 'val_loss', mode = 'min', verbose = 1,
    patience = 30, restore_best_weights = True)
csv_logger = CSVLogger(r'E:\DRIVE\MESTRADO\DISSERTAÇÃO\SALVAMENTO REDE/earlystop02.csv',
    separator = ',', append = False) #VALORES DE TREINAMENTO E VALIDAÇÃO POR ÉPOCA
model_checkpoint = ModelCheckpoint(r'E:\DRIVE\MESTRADO\DISSERTAÇÃO\SALVAMENTO
REDE/earlystop02.hdf5',
    monitor='val_loss', verbose = 1, save_best_only = True) #GERAR O ARQUIVO

#COMPILANDO O MODELO
opt = keras.optimizers.Nadam(learning_rate = 0.001)
model.compile(optimizer = opt, loss = 'mse',
    metrics=['mse', 'mae', 'mape'])

# AJUSTE DE MODELO
history = model.fit(X_train, y_train, epochs=1000, batch_size=10,
    validation_data = (X_test, y_test),callbacks = [model_checkpoint,csv_logger,monitor])

#AVALIAR O CONJUNTO
plt.plot(history.history['mse'])

```

```

plt.plot(history.history['val_mse'])
plt.title('Model MSE')
plt.ylabel('mse')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()

plt.plot(history.history['mae'])
plt.plot(history.history['val_mae'])
plt.title('Model MAE')
plt.ylabel('mae')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()

#FAZER A PREDIÇÃO
pred = model.predict(X_test)

inv_y_pred = np.concatenate((X_test, pred),axis=1)
inv_y_pred = scaler.inverse_transform(inv_y_pred) #INVERTER OS VALORES PREVISTOS PARA O
GRÁFICO
inv_y_pred = inv_y_pred[:,-1]
print(inv_y_pred)

inv_y = np.concatenate((X_test, np.expand_dims(y_test,axis=1)),axis=1)
inv_y = scaler.inverse_transform(inv_y) #INVERTER OS VALORES REAIS PARA O GRÁFICO
inv_y = inv_y[:,-1]

print(inv_y)

#PLOTAR

plt.figure()
x = np.arange(len(pred))
y = np.asarray(inv_y_pred).flatten()
y_test2 = np.asarray(inv_y).flatten()
matrix = np.column_stack((x,y,y_test2))
df = pd.DataFrame(data=matrix)
df.columns = ['id','pred','real']
df.plot(x="id", y= ["pred", "real"], kind="bar")
plt.ylabel('Frequency (Hz)')
plt.xlabel('Data')

```

## RNA LAJES PRÉ-MOLDADAS

```

from tensorflow import keras
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from numpy import loadtxt
from keras.models import Sequential
from keras.layers import Dense
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn import metrics
from keras.callbacks import EarlyStopping, ModelCheckpoint, CSVLogger

#PREPARANDO OS DADOS
dataset_real = loadtxt('bancoxPREMOLDADAS.csv', delimiter=',')
    # Usa essa função para normalizar de 0 a 1
scaler = MinMaxScaler(feature_range=(0, 1))
dataset = scaler.fit_transform(dataset_real)

#DIVIDINDO ENTRE VARIÁVEIS DE ENTRADA (X) E DE SAÍDA (Y)
trainX = dataset[:,0:9] #PRIMEIRAS COLUNAS, DE 0 A N (EX.: 0:8 PARA 7 COLUNAS)
trainY = dataset[:,9] #ÍNDICE QUE SE ENCONTRA A VARIÁVEL DE SAÍDA (EX.: ÍNDICE 8, COLUNA
9)

#SEPARANDO CONJUNTO TESTE E VALIDAÇÃO
X_train, X_test, y_train, y_test = train_test_split(
    trainX, trainY, test_size=0.3, random_state=42) # test_size: PERCENTUAL USADO PARA TESTE

#DEFININDO O MODELO KERAS
#Número de camadas
model = Sequential()
model.add(Dense(5, input_dim=9, activation='relu')) #INPUT_DIM: VARIÁVEIS DE ENTRADA, DENSE: N
CAMADAS INTERMEDIARIAS
model.add(Dense(1, activation='relu')) #CAMADA DE SAÍDA

#EARLYSTOPPING
monitor = EarlyStopping(monitor = 'val_loss', mode = 'min', verbose = 1,
    patience = 30, restore_best_weights = True)
csv_logger = CSVLogger(r'E:\DRIVE\MESTRADO\DISSERTAÇÃO\SALVAMENTO
REDE/earlystopPREMOLDADAS.csv',
    separator = ',', append = False) #VALORES DE TREINAMENTO E VALIDAÇÃO POR ÉPOCA
model_checkpoint = ModelCheckpoint(r'E:\DRIVE\MESTRADO\DISSERTAÇÃO\SALVAMENTO
REDE/earlystopPREMOLDADAS.hdf5',
    monitor='val_loss', verbose = 1, save_best_only = True) #GERAR O ARQUIVO

#COMPILANDO O MODELO
opt = keras.optimizers.Nadam(learning_rate = 0.001)
model.compile(optimizer = opt, loss = 'mse',
    metrics=['mse', 'mae', 'mape'])

# AJUSTE DE MODELO
history = model.fit(X_train, y_train, epochs=3000, batch_size=10,
    validation_data = (X_test, y_test),callbacks = [model_checkpoint,csv_logger,monitor])

#AVALIAR O CONJUNTO
plt.plot(history.history['mse'])
plt.plot(history.history['val_mse'])

```

```

plt.title('Model MSE')
plt.ylabel('mse')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()

plt.plot(history.history['mae'])
plt.plot(history.history['val_mae'])
plt.title('Model MAE')
plt.ylabel('mae')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()

#FAZER A PREDIÇÃO
pred = model.predict(X_test)

inv_y_pred = np.concatenate((X_test, pred),axis=1)
inv_y_pred = scaler.inverse_transform(inv_y_pred) #INVERTER OS VALORES PREVISTOS PARA O
GRÁFICO
inv_y_pred = inv_y_pred[:,-1]
print(inv_y_pred)

inv_y = np.concatenate((X_test, np.expand_dims(y_test,axis=1)),axis=1)
inv_y = scaler.inverse_transform(inv_y) #INVERTER OS VALORES REAIS PARA O GRÁFICO
inv_y = inv_y[:,-1]

print(inv_y)

#PLOTAR

plt.figure()
x = np.arange(len(pred))
y = np.asarray(inv_y_pred).flatten()
y_test2 = np.asarray(inv_y).flatten()
matrix = np.column_stack((x,y,y_test2))
df = pd.DataFrame(data=matrix)
df.columns = ['id','pred','real']
df.plot(x="id", y= ["pred", "real"], kind="bar")
plt.ylabel('Frequency (Hz)')
plt.xlabel('Data')

```



# UPF

UNIVERSIDADE  
DE PASSO FUNDO

UPF Campus I - BR 285, São José  
Passo Fundo - RS - CEP: 99052-900  
(54) 3316 7000 - [www.upf.br](http://www.upf.br)