**Artificial Neural Networks Applied in Mechanical Structural Design**

**Abstrat**

A Inteligência Artificial tem trazido muitas novas abordagens de resolução de problemas à sociedade nos últimos anos. No domínio da Inteligência Artificial, as técnicas de Aprendizagem de Máquinas estão a ser muito bem sucedidas, principalmente devido à sua capacidade de aprender. O principal objetivo deste artigo é a investigação, desenvolvimento e aplicação de técnicas de Inteligência Artificial para a concepção de estruturas mecânicas. O método selecionado baseia-se nas Redes Neurais Artificiais, que permitem a previsão de uma determinada variável com base num dado conjunto de dados. A rede artificial aplicada foi desenvolvida em Python com a biblioteca TensorFlow e a base de dados foi preparada utilizando o software de elementos finitos, Ansys, com linguagem paramétrica para extracção automática de dados. É explorado um estudo de caso clássico de mecânica sólida, compreendendo uma placa com um orifício central sujeita a tensão remota uniaxial. Pretende-se obter a distribuição de tensão para uma placa com um raio de furo entre 25 e 50 mm. Com as Redes Neurais Artificiais, observa-se uma redução substancial do tempo de simulação, sendo, aproximadamente, 79 vezes mais rápido quando comparado com o tempo de solução da abordagem convencional por elementos finitos. A rede neural desenvolvida tem um erro médio relativo de cerca de 4,57%, o que é considerado satisfatório dado que se trata de uma primeira aplicação destas redes neste domínio. Em conclusão, com este trabalho é possível destacar as potenciais vantagens das Redes Neurais Artificiais no cálculo de tensões/esforço em mecânica sólida: menor tempo de resposta, menos recursos computacionais e simplificação do problema, em detrimento de uma menor resolução/atualização do comportamento estrutural. Os procedimentos de optimização e os conceitos de "digital-twin" podem tirar partido destas vantagens, permitindo cálculos quase em tempo real.

1. Introdução

A Inteligência Artificial (IA) está a tornar-se cada vez mais presente na nossa vida quotidiana. Os benefícios que tem trazido à sociedade nos últimos anos são notáveis. A sua utilização em tarefas da vida diária complexa, como a condução autónoma, prova que o ser humano tem cada vez mais confiança neste tipo de soluções. O sucesso dos modelos de IA nos últimos anos está intimamente relacionado com a sua capacidade de aprender. Os modelos com esta propriedade permitem o desenvolvimento de abordagens de Machine Learning (ML), permitindo a sua exploração numa vasta gama de problemas de engenharia. Num programa de computador, é especificado ao computador o que se espera que ele faça quando algum input é apresentado. Por outro lado, nos sistemas ML, em vez de programar as acções, são aprendidas utilizando dados do problema específico que se pretende resolver. Com os avanços nas arquitecturas informáticas e os bons resultados demonstrados pelas tecnologias ML, há um interesse crescente na comunidade científica sobre este tópico. Estão a ser feitos esforços para expandir as áreas de aplicação da IA e para melhorar ainda mais o seu desempenho e para tirar partido das suas capacidades. O objetivo deste artigo é estudar, desenvolver e aplicar técnicas de IA, nomeadamente Redes Neurais Artificiais (ANNs), na concepção mecânica. O estudo de caso específico será o problema clássico de uma placa com furo central sob tensão uniaxial. Desta forma, pretende-se, através das ANNs, prever o campo de tensão da placa e compará-lo com o campo de tensão obtido a partir do software Ansys baseado no método dos elementos finitos. Os termos de comparação são a precisão do modelo, os custos computacionais e o tempo de simulação.

2. Redes Neuronais Artificiais

Os Sistemas Connectionist, mais comummente referidos como ANNs, são modelos ML inspirados no sistema neurológico humano. Talvez as primeiras arquitecturas ANN tenham sido feitas para simular o cérebro humano, tornaram-se ferramentas poderosas para resolver uma grande variedade de problemas. Uma das tarefas mais populares desempenhadas por estes modelos é o reconhecimento de padrões. Para este fim, é dado aos pares input-output da rede, e depois tentará encontrar uma função que se aproxime corretamente das relações reais entre eles. De uma forma simplificada, uma ANN pode ser definida como apresentada em Eq.(1), assumindo f\* como a relação real entre x e y. A função f é definida com a escolha da arquitectura ANN. O processo de aprendizagem determinará os valores de θ [1].

ŷ = f (x,θ) (1)

onde 𝒙 representa os dados de entrada, ŷ os dados de saída previstos, 𝑓 a função aproximada de f\* e 𝜽 os parâmetros de aprendizagem.

A arquitectura ANN mais popular é a rede neural feed-forward, também chamada Multilayer Perceptron (MLP). Nesta arquitectura, a informação flui numa só direção, a partir da camada de entrada, passando pelas camadas ocultas e terminando na camada de saída. Cada nó computacional de uma ANN é chamado de neurónio. A figura 1 mostra um exemplo da arquitetura do MLP.

Quando uma ANN tem mais do que uma camada oculta, pode ser chamada Rede Neural Profunda (DNN), uma vez que a informação é tratada hierarquicamente, em camadas com diferentes profundidades. A partir da Fig.1, pode-se reescrever a definição de ANN, assumindo que a função f em Eq.(2) é a composição de n funções, onde n representa o número de camadas de computação na rede [1],[2].

ŷ =𝑓3(𝑓2(𝑓1( 𝒙,𝜃1),𝜃2),𝜃3) (2)

onde 𝑓𝑖 e 𝜃𝑖 representam a função aproximada e os parâmetros de aprendizagem da camada i, respectivamente.

As camadas apresentadas na Fig.1 estão totalmente ligadas, o que significa que cada neurónio recebe informação de todos os nós da camada anterior, e a sua saída é utilizada de todos os neurónios da camada seguinte.

A estrutura artificial do neurónio é uma tentativa de reproduzir a célula nervosa humana, tal como apresentada na Fig.2. O neurónio recebe informação, processa-a, e depois encaminha-a para os nós adjacentes. O processo de transmissão de informação é chamado de sinapse. No neurónio artificial, a informação é processada em duas fases. A primeira fase é uma soma ponderada de todos os dados de entrada e a segunda é executada por uma função de activação, que decide se o neurónio é activado ou não, ou seja, se a informação passará para os neurónios adjacentes [3]. A função de activação (φ) mapeia a soma ponderada de ]-∞; +∞[ intervalo para o domínio desejado. Uma das funções de activação mais populares é a Unidade Linear Retificada (ReLU), que tem um domínio de saída de [0, +∞[, ativando o neurónio apenas quando a entrada é superior a 0 [4]. No processo de aprendizagem, ao nível do neurónio, os parâmetros de aprendizagem são os pesos das conexões de entrada e o viés, definido como b. O viés é um parâmetro adicional para melhorar a flexibilidade das redes neuronais.

𝑦=𝑅𝑒𝐿𝑈(𝑤1⋅𝑥1+𝑤2⋅𝑥2+𝑤3⋅𝑥3+𝑏) (3)

onde 𝑥𝑖 e w𝑖 são os dados de entrada do neurónio i da camada anterior e o seu peso associado, b é o enviesamento e 𝑦 representa a informação processada. A função ReLU é definida em Eq.(4).

𝑅𝑒𝐿𝑈(𝑥)=max(0,𝑥)={0,𝑖𝑓 𝑥<0𝑥,𝑖𝑓 𝑥≥0 (4)

O processo de aprendizagem é um loop, começando com todos os pesos definidos com valores aleatórios, sendo ajustados em cada iteração. Cada iteração tem duas fases: a propagação para a frente, onde os dados de entrada (x) são utilizados para prever o valor de saída (ŷ); e a atualização dos pesos, utilizando o algoritmo de retropropagação para calcular o gradiente de erro para cada camada, propagando-o a partir da saída para as outras camadas [5]. Este processo é determinado por dois elementos principais: a função de perda e o otimizador. A função de perda define uma medida do sucesso da rede na sua tarefa durante as iterações de aprendizagem. Utiliza os resultados previstos (ŷ) pela rede em cada etapa e os valores reais (y) para calcular o erro. Uma das funções de perda mais utilizadas nos modelos de regressão é o Erro Quadrático Médio (MSE) definido na equação Eq.(5).

MSE(y,ŷ) = 1𝑛Σ(𝑦𝑖-ŷ𝑖)2𝑛𝑖=1 (5)

onde n é o número de amostras de dados. O principal objetivo é minimizar a função de perda. Utilizando o gradiente de erro, a ANN deve convergir para o seu mínimo. O otimizador define a forma como os pesos na rede são atualizados com base no gradiente calculado com o algoritmo de retropropagação. Os otimizadores implementam alguma variante de descida de gradiente estocástico (SGD). Alguns dos otimizadores mais populares para problemas de regressão são RMSProp e Adam [1],[2].