



Proposta de utilização de redes neurais *feedforward* multicamadas para a otimização de padrões de recarga do combustível em um reator PWR

Aruquipa W., Oliveira E., Silva C.A.M., Pereira C.

*Departamento de Eng. Nuclear- UFMG, PCA1—Anexo Engenharia,
Av. Antonio Carlos, 6627 Pampulha 31270-901, Belo Horizonte, MG, Brasil
wilmer.fcfn.fis@gmail.com; claubia@nuclear.ufmg.br*

RESUMO

O gerenciamento de recarga de combustível em um reator de potência tem como principal foco o padrão de recarga no núcleo de modo a alcançar melhor rendimento no ciclo observando todos os parâmetros de segurança adotados. Neste artigo apresentamos a estratégia de um algoritmo baseado em redes neurais artificiais que poderia ser usado para otimizar a recarga de combustível num reator nuclear. A ideia é apresentar uma metodologia baseada em rede neural *feedforward* multicamadas baseada em neurônios multi-valorados, que poderá ser usada para desenvolver uma metodologia capaz de escolher as melhores combinações que satisfaçam o fator de pico de potência radial e maximizem o fator de multiplicação efetivo no início do ciclo, e também satisfaçam a relação de potência crítica mínima e taxa máxima de geração de calor no final do ciclo.

Palavras-chave: Reator PWR, Redes neurais artificiais, Carga de combustível

ABSTRACT

The management of refueling in a power reactor has as its main focus the refueling pattern in the core in order to achieve better performance in the cycle observing all the safety parameters adopted. In this paper we present the strategy of an algorithm based on multi-layered *feedforward* neural network based on multi-valued neurons that will be used to develop a methodology capable of choosing the best combinations that satisfy the radial power peak factor and maximize the effective multiplication factor at the beginning of the cycle and also satisfy the ratio of minimum critical power and maximum rate of heat generation in the end of the cycle.

Keywords: Reactor PWR, Artificial neural networks, Fuel load

1. INTRODUÇÃO

O projeto de recarga para um ciclo de operação de um reator nuclear é um problema complexo. O problema consiste em como distribuir os N elementos combustíveis, compostos de grupos diferentes segundo o tipo de sistema de recarga escolhido, substituição de $1/3$ ou $1/4$ do combustível queimado, e redistribuir estes N elementos de modo a obter as melhores condições de funcionamento durante o ciclo, observando todos os parâmetros de segurança determinados no projeto do reator.

Na literatura, vários trabalhos são mencionados onde esse problema é resolvido aplicando-se sistemas baseados em parâmetros de segurança [1], algoritmos genéticos [2], redes neurais artificiais (RNA) [3], busca tabu [4], *simulated annealing* [5], etc. Um aspecto em comum em muitos desses trabalhos é que eles contemplam a colocação dos elementos combustíveis mantendo uma simetria entre as diferentes áreas do reator. Algumas análises são realizadas usando um oitavo do núcleo, e acrescentam que seria suficiente colocar os elementos de combustível de características semelhantes nas posições correspondentes nas outras 7 partes do reator. As complicações surgem quando não há elementos combustíveis de características similares em quantidade suficiente para serem colocados em tais posições.

O ideal seria fazer o estudo usando o núcleo inteiro, mas isso tornaria o processo muito lento, porque isso corresponderia a N^N possibilidades para cada recarga e os parâmetros deveriam ser validados em um código de análise de núcleo e comparar os resultados obtidos e determinar qual a distribuição que cumpre com todos os requisitos deveria ser avaliado com um simulador de reator para verificar se ele está em conformidade com as restrições. Um ponto intermediário é usar uma simetria de $1/4$ porque a simulação levaria menos tempo para fazer a avaliação. Outra característica encontrada em alguns trabalhos relatados na literatura é que a otimização parcial é feita, isto é, que se busca maximizar uma variável, mas outras igualmente importantes não são observadas [3].

Neste trabalho, ainda de forma teórica e propositiva, a ideia principal é estudar a rede neural tipo *feedforward* de multicamadas baseada em neurônios multi-valorados [6, 10, 12] e posteriormente avaliar a possibilidade de sua utilização para desenvolver uma estratégia de recarga de um reator PWR. A ideia posterior principal será, após sua implementação, é treinar a RNA com

informações sobre o desempenho de um conjunto de configurações espaciais para os conjuntos de combustível, utilizando-as posteriormente para gerar novas configurações que serão avaliadas de acordo com o desempenho dos parâmetros obtidos através das simulações com códigos neutrônicos.

Este trabalho, de cunho teórico e computacional, baseia-se na implementação de uma rede neural tipo *feedforward* de multicamadas baseada em neurônios multi-valorados foi analisada e estudada. A implementação da rede foi feita em Python 3.7.1, e considerando que esta rede foi proposta pela primeira vez em 2014 [10], entende-se que sua aplicação à área da energia nuclear será bastante inovadora. Baseado nas referências, essa rede neural promete um tempo de cálculo mais curto, além de maior eficiência no reconhecimento de padrões [6, 10, 12].

2. REDES NEURAIS ARTIFICIAIS

As redes neurais artificiais (RNAs) são modelos baseados na estrutura do cérebro humano e são utilizados para problemas complicados de reconhecimento, agrupamento, classificação e simulação de padrões. Foi demonstrado que RNAs são aproximações de funções universais que são capazes de mapear qualquer função não linear complicada [7]. A RNA pode aprender inteligentemente essas funções através de um processo de treinamento. A capacidade dos RNAs mapearem um conjunto de dados de entrada/saída com uma faixa aceitável de erros os torna ferramentas úteis para modelagem de processos naturais. Como na maioria dos modelos baseados em dados, os RNAs não geram uma relação paramétrica entre variáveis independentes e dependentes. Em vez disso, eles fornecem uma relação entre dados de entrada e saída por meio de um processo de treinamento para aproximar qualquer função contínua com uma precisão específica. A abordagem de RNA é uma maneira eficaz e eficiente de modelar problemas dependentes de dados em situações em que o conhecimento explícito do subprocesso físico interno não é necessário ou ainda não foi descoberto [8].

2.1. Multi-layered feedforward neural network based on multi-valued neurons

O trabalho desenvolvido por Ortiz e Requena [9] mostra uma metodologia baseada em redes neurais recorrentes de múltiplos estados, e é usada para obter o projeto de recarga de combustível para um reator BWR; inspirado por este método, este trabalho tem como objetivo propor uma metodologia baseado no uso da rede neural *feedforward* multicamadas baseadas em neurônios

multi-valorados (Multilayer Neural Network Based On Multi-Valued Neurons, MLMVN) [6, 10, 12].

Um neurônio de valor múltiplo (Multi-Valued Neuron, MVN) é baseado nos princípios da lógica de limiar de múltiplos valores no campo dos números complexos. As propriedades mais importantes do MVN são: os pesos são números complexos, entradas e saídas codificadas pelas raízes k^{th} da unidade e a função de ativação, que mapeia o plano complexo no círculo unitário [10]. O aprendizado MVN é reduzido ao movimento ao longo do círculo unitário, é baseado em uma regra de correção de erro linear simples e não requer uma derivada. É mostrado que, usando uma arquitetura de rede neural *feedforward multilayer* tradicional (Multilayer Feedforward Neural Network, MLF) e a alta funcionalidade do MVN, é possível obter uma nova rede neural poderosa. Seu treinamento não requer uma derivada da função de ativação e sua funcionalidade é maior do que a funcionalidade MLF que contém o mesmo número de camadas e neurônios [6, 10, 12].

2.2. Neurônio de Valores Múltiplos (MVN)

2.2.1. MVN Discreto

Um valor discreto MVN é um mapeamento de função de uma entrada de n características para uma única saída. Este mapeamento é descrito por uma função de valor múltiplo de n variáveis $f(x_1, \dots, x_n)$ com $n + 1$ pesos complexos como parâmetros [10]:

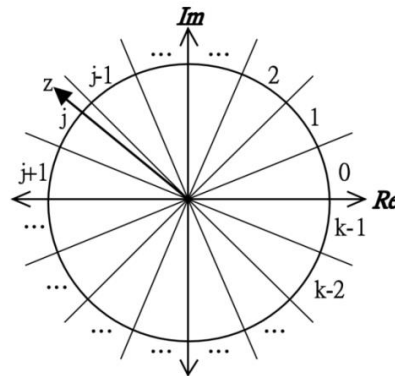
$$f(x_1, \dots, x_n) = P(\omega_0 + \omega_1 x_1 + \dots + \omega_n x_n) \quad (1)$$

Onde x_1, x_n são as características de uma instância, da qual a função, $\omega_0, \omega_1, \omega_n$ são os pesos, eles são as raízes k^{th} da unidade. $\varepsilon^j = \exp(i2\pi j/k)$, $j: 0, 1, \dots, k - 1$, i é a unidade imaginária. P é a função de ativação do neurônio:

$$P(z) = \exp\left(\frac{i2\pi j}{k}\right), \text{ si } \frac{2\pi j}{k} \leq \arg(z) < \frac{2\pi(j+1)}{k} \quad (2)$$

onde $z = \omega_0 + \omega_1 x_1 + \dots + \omega_n x_n$ é a soma ponderada. A Eq. 2 é ilustrada na Figura. 1.

Figura 1: Interpretação geométrica da função de ativação discreta MVN [10]



A aprendizagem MVN é reduzido ao movimento ao longo do círculo unitário e é livre de derivação. O movimento é determinado pelo erro que é a diferença entre a saída desejada e a real. A regra de correção dos erros de aprendizagem e o algoritmo de aprendizado correspondente para o valor discreto MVN foram descritos em [6] e modificados em [10].

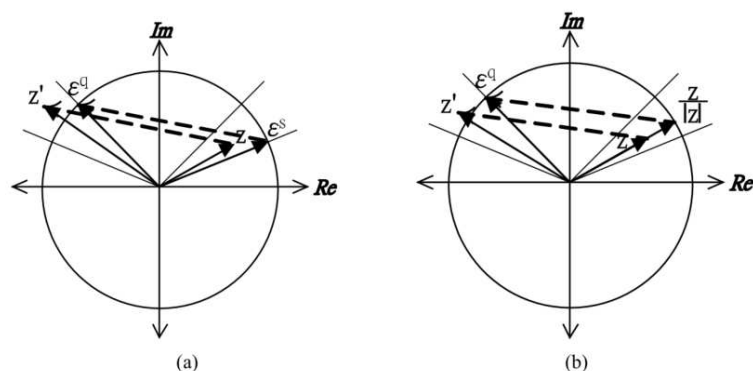
2.2.2 MVN contínuo

A Eq. 2 da função de ativação é descontínua. Esta função pode ser modificada e generalizada para o caso contínuo da seguinte maneira. Quando $k \rightarrow \infty$ na Eq. 2, o valor do ângulo do setor (veja a Fig.1) se aproximará de zero. A função de ativação é transformada da seguinte maneira [6]:

$$P(z) = \exp(i \arg(z)) = \frac{z}{|z|} \quad (3)$$

onde z é a soma ponderada, $\arg(z)$ é o argumento do número complexo z . A Eq. 3 da função de ativação mapeia a soma ponderada para o círculo unitário completo (veja a Fig. 2 (b)). A Eq. 2 é atribuída apenas aos subconjuntos discretos dos pontos pertencentes ao círculo unitário. As Eq. 2 e 3 não são diferenciáveis, mas sua diferenciabilidade não é necessária para a aprendizagem MVN.

Figura 2: Interpretação geométrica da regra de aprendizagem da MVN: (a) MVN de valor discreto; e (b) valor MVNs contínuos [10].



O MLMVN é uma rede neural multicamadas com potência padrão, onde as saídas de neurônios na camada anterior são conectadas às entradas correspondentes de neurônios na próxima camada. A rede contém uma camada de entrada, camadas ocultas e uma camada de saída.

3. METODOLOGIA

Partindo da metodologia usada em [9], para que funcione adequadamente, o MLMVN requer duas especificações, uma função de energia e uma regra de transição de estados entre os neurônios.

A regra da transição de estados entre os neurônios é a maneira pela qual dois ou mais neurônios trocam seus estados. Neste caso, é feito escolhendo dois neurônios aleatoriamente e seus estados são trocados; isto é equivalente a trocar 2 elementos combustíveis de diferentes canais. Além disso, para manter a simetria em cada quarta parte do núcleo, os neurônios selecionados para a troca de estado podem ser apenas um quarto do núcleo e, conseqüentemente, os neurônios correspondentes dos outros quartos do núcleo também trocam seus estados [9].

A função de energia quantifica a energia total da rede quando todos os neurônios têm um estado associado. Durante o processo de troca de estados, é desejável que o valor da função de energia seja diminuído e estabilizado ao mínimo. No nosso caso, queremos maximizar o valor de k -eff no início do ciclo por meio da regra de transição de estado. É por isso que é conveniente definir uma função de energia adequada. Em [9] é usada a seguinte função que pode ser derivada da Equação de Difusão de Nêutrons em dois grupos de energia

$$F = \frac{\sum_{i=1}^{111} \phi_{2i} \left[\frac{\nu_1 \Sigma_{f1}^i \Sigma_{2a}^i}{\Sigma_{1 \rightarrow 2}^i} + \nu_2 \Sigma_{f2}^i \right] + \Delta N_{jk}}{\sum_{i=1}^{111} \phi_{2i} \Sigma_{2a}^i \left[\frac{\Sigma_{1a}^i}{\Sigma_{1 \rightarrow 2}^i} + 1 \right] + \Delta D_{jk}} \quad (3)$$

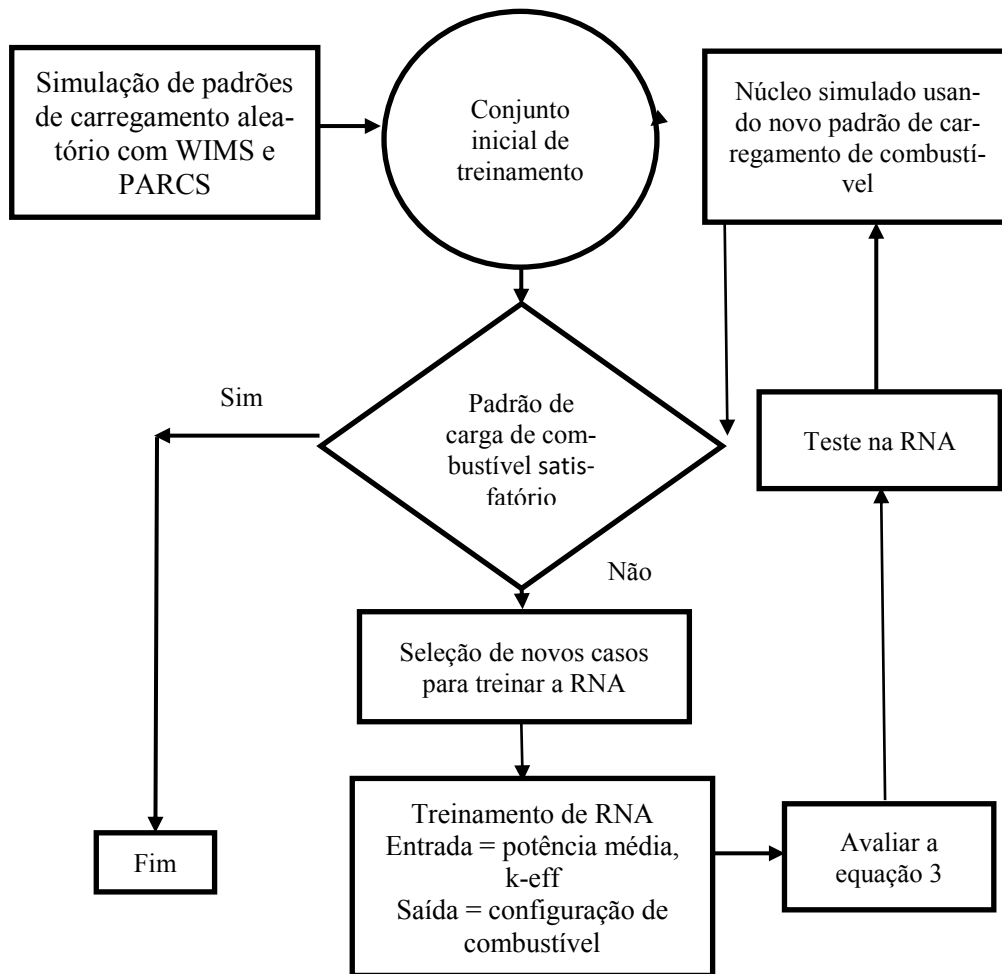
onde Σ_{1a} y Σ_{2a} são as seções de choque de absorção efetiva, $\Sigma_{1 \rightarrow 2}$ a de espalhamento e Σ_{f1} e Σ_{f2} a de fissão para cada grupo de energia considerado, sendo 1 correspondente aos nêutrons rápidos e 2 aos térmicos; o fluxo térmico de nêutrons também está envolvido (ϕ_{2i}). O índice i é aplicado a cada um dos elementos combustíveis, o que significa que as seções de choque envolvidas devem ser para o elemento combustível. ΔN_{jk} e ΔD_{jk} são calculados por meio de:

$$\Delta N_{jk} = \left[\left(\frac{\nu_1 \Sigma_{f1}^j \Sigma_{2a}^j}{\Sigma_{1 \rightarrow 2}^j} + \nu_2 \Sigma_{f2}^j \right) \phi_k + \left(\frac{\nu_1 \Sigma_{f1}^k \Sigma_{2a}^k}{\Sigma_{1 \rightarrow 2}^k} + \nu_2 \Sigma_{f2}^k \right) \phi_{ij} \right] - \left[\left(\frac{\nu_1 \Sigma_{f1}^j \Sigma_{2a}^j}{\Sigma_{1 \rightarrow 2}^j} + \nu_2 \Sigma_{f2}^j \right) \phi_j + \left(\frac{\nu_1 \Sigma_{f1}^k \Sigma_{2a}^k}{\Sigma_{1 \rightarrow 2}^k} + \nu_2 \Sigma_{f2}^k \right) \phi_k \right] \quad (4)$$

$$\Delta D_{jk} = \left\{ \left[\phi_{2k} \Sigma_{2a}^j \left(\frac{\Sigma_{1a}^j}{\Sigma_{1 \rightarrow 2}^j} + 1 \right) \right] + \left[\phi_{2k} \Sigma_{2a}^k \left(\frac{\Sigma_{1a}^k}{\Sigma_{1 \rightarrow 2}^k} + 1 \right) \right] \right\} - \left\{ \phi_{2j} \Sigma_{2a}^j \left(\frac{\Sigma_{1a}^j}{\Sigma_{1 \rightarrow 2}^j} + 1 \right) + \left(\phi_{2j} \Sigma_{2a}^k \left(\frac{\Sigma_{1a}^k}{\Sigma_{1 \rightarrow 2}^k} + 1 \right) \right) \right\} \quad (5)$$

Nessas expressões, os dois primeiros termos apresentam uma contribuição para o valor de F quando o neurônio j troca seu estado pelo neurônio k; por outro lado, os dois últimos aditivos subtraem a contribuição dos neurônios quando associaram os estados k e j, respectivamente. Devido a estas duas últimas equações, não é necessário avaliar a expressão de F para conhecer a energia do novo estado neuronal. É suficiente que $\Delta N_{jk} > \Delta D_{jk}$ para F aumente em valor; e o oposto se $\Delta N_{jk} < \Delta D_{jk}$ F diminuirá seu valor. Tomando $k_{\text{eff}} = F$, estaremos maximizando seu valor.

Figura 3: Diagrama de blocos do algoritmo implementado.



No algoritmo descrito na Figura 3, inicialmente é gerado um grupo de padrões de carga aleatório para conjuntos de combustível. Cada uma dessas configurações para o combustível, é simulada usando os códigos PARCS [11] e WIMS [13]. Os dados obtidos sobre o perfil radial do fator de potência médio, juntamente com a configuração espacial associada dos conjuntos de combustível, formam a base do padrão para o treinamento de RNA. Após o treinamento do RNA, uma nova configuração é selecionada com os dados de potência média e k-eff como entrada e a configuração da carga de combustível como saída, para depois avaliá-lo na equação 3. Após avaliar os parâmetros, esses valores são testados no RNA e, finalmente, o núcleo é simulado com o novo padrão de carga.

Este trabalho, ao contrário de outros baseados em RNAs, utiliza o MLMVN, que será implementado no método proposto por ORTIZ [9]. O código usado para o RNA é uma adaptação do código desenvolvido por Hlaváček, disponível em [10,12].

4. RESULTADOS E CONCLUSÃO

Um novo tipo de rede neural está sendo estudado a fim de aplicá-lo para obter uma configuração ótima dos elementos combustível em um reator PWR. O objetivo é alcançar maior eficiência na obtenção da potência desejável dos elementos combustíveis, sem comprometer a segurança do reator. Esta proposta de rede promete um modelo de cálculo mais fácil, porque os pesos são números complexos. A ideia principal é usar redes neurais *feedforward* multicamadas baseadas em neurônios multivalorados. Para verificar o algoritmo, será considerado um reator PWR e se simulará $\frac{1}{4}$ do núcleo. Neste trabalho o algoritmo proposto por Hlaváček foi escrito e compilado em Python 3.7.1., e do ponto de vista numérico, não apresentou instabilidades. Cabe ressaltar que a estabilidade do método do ponto de vista de sistemas dinâmicos ainda não foi testada. O tempo de treinamento em testes de reconhecimento de padrões para a rede estudada é mais rápido que para os RNAs clássicos, em simulação de problemas clássicos tais como classificação de imagem, reconhecimento, etc. No campo dos reatores nucleares ainda não foi avaliado, mas acreditamos que o tempo de treinamento será mais rápido e, em outras áreas de estudo, elas foram bastante eficientes. Quanto à confiabilidade dos resultados a serem obtidos, como esta é uma RNA do tipo *feedforward*, isso dependerá muito dos dados de treinamento.

Para validar e testar a funcionalidade da rede neural, o código neutrônico PARCS será usado para calcular os fatores de pico de potência e o fator de multiplicação de nêutrons; as seções de choque serão geradas com o código de cálculo de célula WIMS5D. Pretende-se utilizar os valores dos fatores de pico de potência e o fator de multiplicação da saída do código PARCS como dados de entrada na rede neural para obter a configuração dos elementos combustíveis. Espera-se encontrar diversas configurações para o núcleo, apresentando melhores características que a configuração de referência adotada, o que indicaria a viabilidade da metodologia desenvolvida.

AGRADECIMENTOS

Os autores agradecem à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) que direta ou indiretamente contribuíram para a pesquisa e aos patrocinadores e voluntários doadores pelo apoio à realização deste evento.

REFERÊNCIAS

- [1] H. G. KIM, S. H. CHANG, AND B. H. LEE, Optimal Fuel Loading Pattern Design Using an Artificial Neural Network and a Fuzzy Rule-Based System, **Nuclear Science and Engineering**, v. 115, p. 152–163, 1993.
- [2] R. CARMONA, C. MARTIN-DEL-CAMPO, I. OROPEZA, AND J.-L. FRANOIS, Genetic algorithm approach for radial fuel lattice optimization in BWRs, **American Nuclear Society**, vol. 1, p. 323-333, 2009.
- [3] E. F. FARIA AND C. PEREIRA, Nuclear fuel loading pattern optimisation using a neural network, **Annals of Nuclear Energy**, v. 30, p. 603–613, 2003.
- [4] C. LIN, J.-I. YANG, K.-J. LIN, AND Z.-D. WANG, Pressurized Water Reactor Loading Pattern Design Using the Simple Tabu Search, **Nuclear Science and Engineering**, v. 129, p. 61–71, 1998.
- [5] T. ŠMUC, D. PEVEC, AND B. PETROVIC, Annealing strategies for loading pattern optimization, **Annals of nuclear energy**, v. 21, p. 325–336, 1994.
- [6] I. AIZENBERG AND C. MORAGA, Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm, **Soft Computing**, v. 11, p. 169–183, 2007.
- [7] E. M. AZOFF, **Neural network time series forecasting of financial markets**. John Wiley & Sons, Inc., 1994.
- [8] M. T. HAGAN, H. B. DEMUTH, M. H. BEALE, AND O. DE JESUS, **Neural network design**, v. 20. Pws Pub. Boston, 1996.
- [9] J. J. ORTIZ AND I. REQUENA, Using a multi-state recurrent neural network to optimize loading patterns in BWRs, **Annals of Nuclear Energy**, v. 31, p. 789–803, 2004.
- [10] M. HLAVÁČEK, **Multilayer feedforward neural networks based on multi-valued neurons**, PhD Thesis, Masarykova univerzita, Fakulta informatiky, 2014.
- [11] DOWNAR, T. et al., PARCS v2.7 U.S. NRC Core Neutronics Simulator User Manual, **School of Nuclear Engineering Purdue University**, 2006.
- [12] M. HLAVÁČEK, Machine learning tool: simulator of multi-layered neural network (like MLP) with multi-valued neurons. - **MiraHead/mlmvmn**. 2018.
- [13] WIMSD-5B, 2007 - RSICC Computer Code Collection WIMSD-5B.12 – Deterministic Code System for Reactor-Lattice Calculations – **Oak Ridg National Laboratory**, 2007.