Artificial-Intelligence based DC-DC Converter Efficiency Modelling and Parameters Optimization

Fanghao Tian, Diego Bernal Cobaleda and Wilmar Martinez
Electrical Engineering Department (ESAT)
KU Leuven - Energyville
Diepenbeek - Genk, Belgium
fanghao.tian@kuleuven.be

Keywords

≪DC-DC converter≫, ≪Design optimization≫, ≪Neural network≫, ≪Artificial intelligence≫

Abstract

This paper proposes a modeling and parameter design method for DC-DC converters based on artificial intelligence (AI). Initially, a database of switching losses is constructed using Spice simulation data with a single fixed semiconductor switch. Next, an artificial neural network (ANN) is trained by the database. Then Transfer Learning (TL) is implemented to train other ANNs for other switches with much less training data needed. Finally, under the restrictions of current and voltage ripples, a heuristic optimization algorithm is used to obtain the most efficient and optimal design. The results show that the ANN models give precise estimates of the converter properties.

Introduction

DC-DC power converters are widely used in many applications, such as electric vehicles [1], electric aircraft [2], and solar photovoltaic systems [3]. Additonally, the rising demand of these applications presents new challenges for power converter design, specifically on efficiency, volume, and mass. As a result, design automation has emerged as a novel research area in power electronics, with the goal of using artificial intelligence (AI) approaches to automate and accelerate the design process. However, the design of power converters is complicated and includes topology design, parameter selection, semiconductor and inductor modelling, power loss calculation, and optimization. Consequently, the design of power converters demands extensive simulation resources and time. However, AI has the ability to expedite the design process and reduce reliance on humans, hence enabling design automation.

Authors in [4] proposed an artificial neural network (ANN) based model named Mag-net, which is trained by data from practical experiment measurements, to calculate the magnetic core loss in a fast way. Furthermore, Transfer learning (TL), which is an idea of training a new ANN with much fewer data by adopting a trained ANN in a similar domain [5], is implemented to expand their database on more magnetic materials quickly [6]. Additionally, an ANN-based inductor evaluation model called AI-Mag, which is trained by data from Finite Element Methods (FEM) simulation, is proposed for accelerate the inductor design [7]. Besides, AI is also utilized to optimize the circuit parameters of power converters [8], and the ANN model is also trained to calculate efficiency under different combinations of frequency, inductance and capacitance values with the constraints of current and voltage ripples.

This paper is organized as follows: The data generation and neural network algorithm are explained in the first section. Following this, the first ANN is trained, and TL is utilized in training other ANNs with much less training data. Finally, based on all the ANN models, the particle swarm optimization (PSO) algorithm is used to optimize the efficiency, followed by results and conclusion.

Data Generation

Semiconductor switches are essential components in power converters, which usually work in high frequency and have unavoidable losses. In addition, It is difficult to derive an accurate mathematical power loss model of the switches under different working conditions. However, numerical simulation software such as LT Spice helps to evaluate the performance of circuits accurately. In this paper, a DC-DC synchronous buck converter is set as an example Fig. 1.

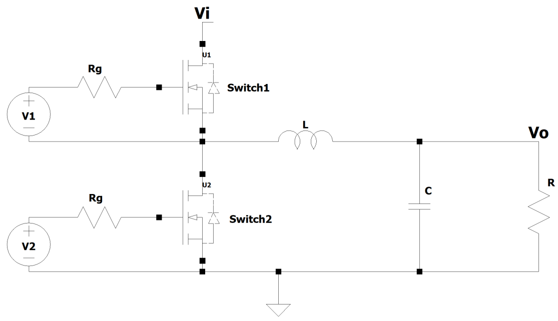


Fig. 1: Schematic of buck converter in LT Spice

The simulation parameters are shown in Table I. An input voltage of 30 $[V]$ is converted to 12 $[V]$ with an output power of 60 $[W]$. Frequency, inductance, and capacitance values are set as variables with a range. 20 frequencies, 50 inductance values, and 10 capacitance values are chosen to form $20*50*10 = 10000$ combinations of parameters for one specific type of switch. By simulating each case, measurement results of current and voltage on both sides are collected, based on which the efficiency, current ripple and voltage ripple are calculated to construct the training dataset.

Table I: Parameters setting for buck converter

Parameters	Values
Input Voltage (V)	30
Output Voltage (V)	12
Power (W)	60
Frequency(kHz)	10-1000
Inductance(μH)	10-1510
Capacitance(nF)	50-5050

In the simulation, Python is used to automate the data generation process. A list of different switches is created with their parameters, including component name, maximal drain-to-source voltage and current, rising and falling time. The external gate resistance is set as $5[\Omega]$, the dead time is set as $10[ns]$, and the rising and falling time are obtained based on the information from datasheets [10].

ANN Design and training

ANN is extensively used for supervised learning on a variety of classification and regression tasks, where labeled data is utilized to train the ANN, allowing it to identify the relationships between inputs and outputs. The fundamental ANN consists of layers of neural nodes. Nodes in adjacent layers are interconnected, forming a network. Every node is a function of all nodes from previous layer, using a weighted linear function and a nonlinear activation function. An example of ANN with 2 hidden layers is illustrated in Fig. 2.

For each node, the weighted linear transfer function and the nonlinear activation function are applied as shown in (1) and (2). The chosen activation function is Rectified Linear Unit (ReLU) function.

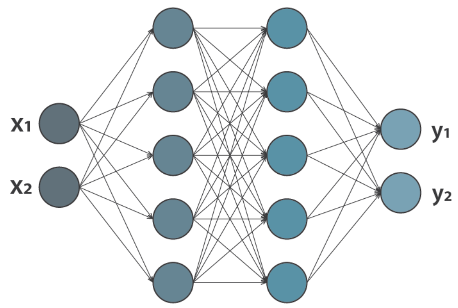


Fig. 2: Structure of an ANN

$$a = \mathbf{wx} + b \tag{1}$$

$$y = max(0, a) \tag{2}$$

Where $\mathbf{x}$ is the vector of all values from the previous layer, $\mathbf{w}$ is the weight parameters from the nodes of previous layer to the current node, and b is the bias. In the ReLU function, positive a is kept while negative a is set as 0.

The ANN used to model the DC-DC buck converter is constructed as shown in Fig. 3. The ANN is trained with frequency, capacitance, and inductance data to model the outputs of efficiency, current ripple, and voltage ripple.

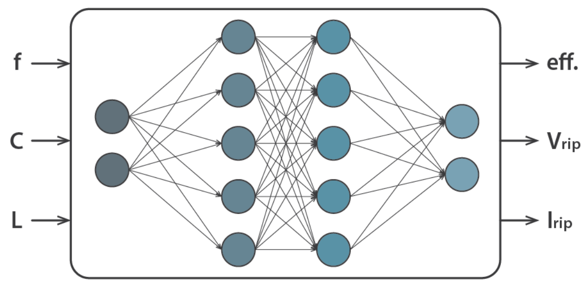


Fig. 3: The ANN model for modelling buck converter

In order to avoid training errors caused by parameters range differences, a logarithmic or linear normalization procedure is needed for the data as shown in (3) and (4) respectively.

$$y_{log} = log_{10}x \tag{3}$$

$$y_{linear} = \frac{x}{x_{max} - x_{min}} \tag{4}$$

Both of logarithmic and linear normalization are implemented on the switching frequency of the converter, and only linear normalization is implemented on the inductance and the capacitance values.

A benchmark switch is selected for the training of the benchmark ANN model. 10,000 sets of data are randomly divided into a training dataset (80%), validation dataset (10%), and testing dataset (10%). To obtain a precise ANN model, various ANN sizes with varying numbers of layers and nodes are evaluated. Eventually, a 4-layer ANN with 128 nodes per layer is selected for training because it has the least training error.

TL for other switches

TL is a technique for training a new ANN with significantly less data by adopting a previously trained ANN in a similar domain. The benchmark ANN model cannot be directly applied to other switches.

However, various switches share similar characteristics. As a result, TL can work as an effective way for training new ANN models for other switches, thereby constructing a database of ANN models for other switches that provide data for optimizing power converter design parameters. The workflow of TL is depicted in Fig. 4.

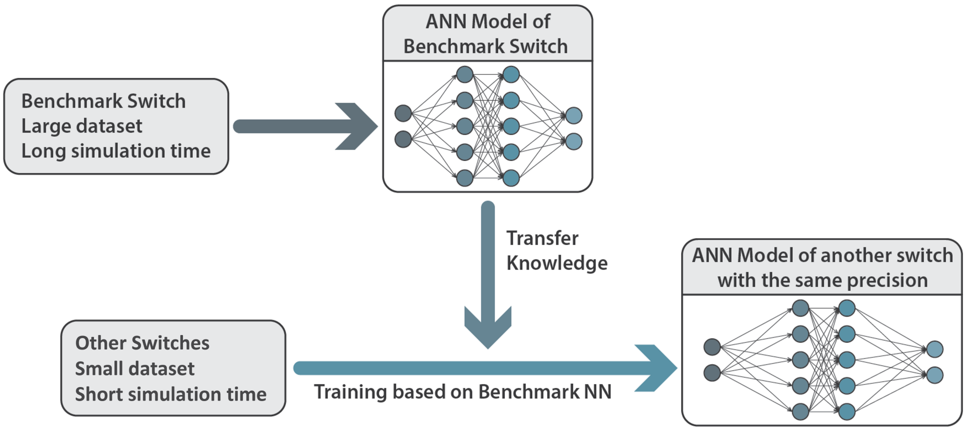


Fig. 4: Workflow of TL

Once the ANN model of the benchmark switch has been obtained, the data for another switch is also collected using a significantly smaller sample size of 2,000, which is only 20% of the size of the benchmark data set. Likewise, the 2000 data points are sorted into three categories. By using the benchmark ANN as the initial state, knowledge from the benchmark ANN model is transferred into the new ANN model, resulting in the need for far fewer data points to reach the same level of precision in the new ANN model as the benchmark ANN.

Parameters optimization by PSO Algorithm

Once the ANN models for several switches have been constructed. The efficiency and ripples can be easily calculated under various parameter combinations. PSO, a classic optimization algorithm, is applied to optimize the input parameters for maximizing the efficiency with satisfying ripples constraints. The optimization problem can be described as (5).

$$\begin{cases} \eta = f(switch, f, C, L) \\ switch \in switch\ list \\ Irip < Irip_{max}, Vrip < Vrip_{max} \end{cases} \tag{5}$$

where η is the efficiency and the optimization objective is to maximize the efficiency.

PSO is a classic bio-inspired optimization algorithm that imitates the swarming behaviors of birds flocking [10]. A group of searching agents work at the same time. After initializing the positions and velocity of the particle swarm, the particle with the best performance on the objective function is defined as global best $gbest$. In the meantime, the best position of the single particle during the whole iteration process is defined as particle best $pbest$. Each particle updates its speed and position as follows:

$$\begin{cases} v = v + c_1 \times r_1 \times (pbest - x) + c_2 \times r_2 \times (gbest - x) \\ x = x + v \end{cases} \tag{6}$$

where v is the updating speed and x is the position. c_1 and c_2 are the learning rate for individual and global updates. Additionally, r_1 and r_2 are two random number between $[0, 1]$, which increases the randomness and enhances the searching ability of the swarm. During the iteration process, the updating procedure is repeated until the optimal position converges to a stable value.

Finally, the whole design and optimization procedures is shown as Fig. 5

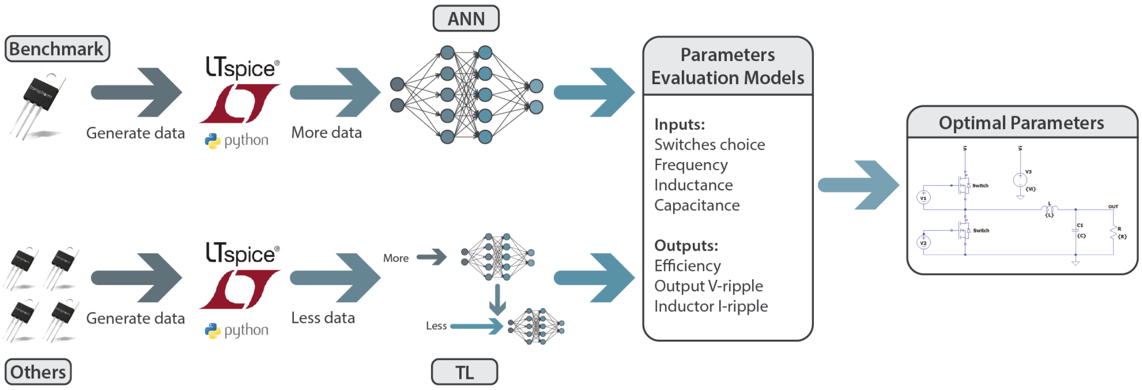
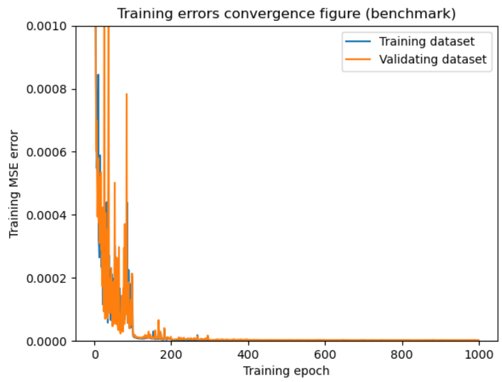


Fig. 5: Design and optimization workflow

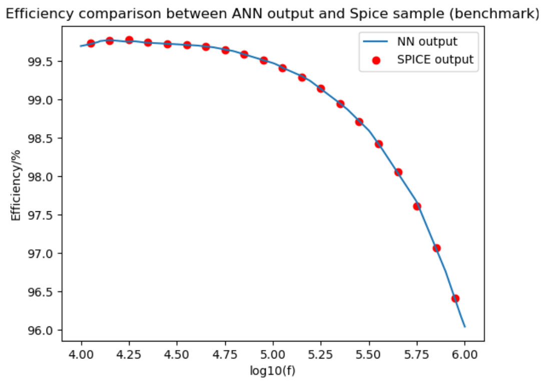
Results

ANN training

The benchmark ANN was initially trained for 1000 epochs. As demonstrated in Fig. 6(a), the mean squared errors (MSE) of the training and validation datasets converged to small values. The decreasing errors in the validation dataset indicate that the ANN is not over-fitted. The error on the test dataset including 1000 sets of data is 2.67×10^{-6}. As a result, the trained ANN is effective at simulating efficiency and ripples. Fig. 6(b) depicts a comparison between the efficiency of the benchmark ANN and the Spice simulation. The curve shows the efficiency changes by frequency when capacitance is $2.5[\mu F]$ and inductance is $1[mH]$. The frequency varies from 10 [kHz] to 1 [MHz] on the logistic scale. The data represented by red dots, which were not part of the training data, are collected from Spice simulations. As depicted in Fig. 6, the benchmark ANN model is well trained and can compute the efficiency precisely under various scenarios.



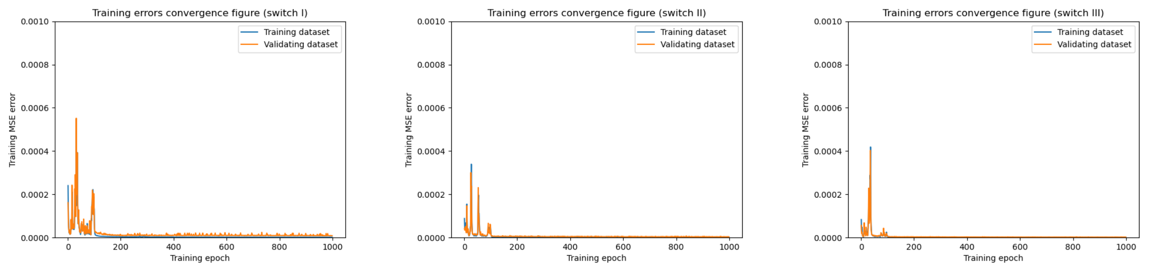
(a)



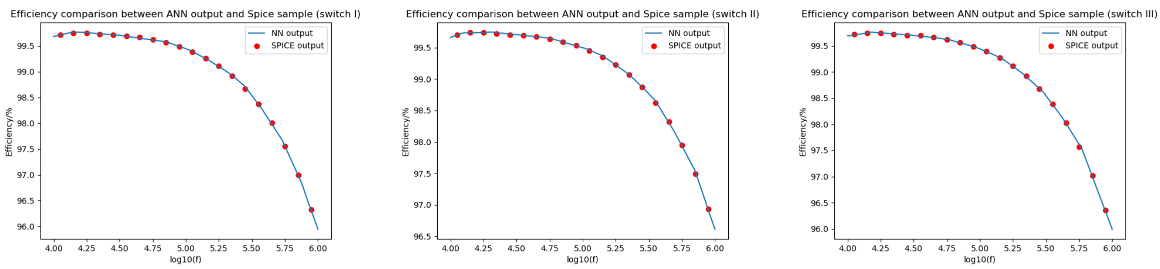
(b)

Fig. 6: Training loss (a) and efficiency validation (b) for the benchmark switch

After obtaining the benchmark ANN model, data from other switches is collected and used to train the other ANN models. On the basis of TL theory, the benchmark ANN can be utilized to efficiently train other models. For the generation of ANN models for other switches, only a modest database, whose size is only 10% of the benchmark database, is required. Fig. 7 shows the results of ANN models for three additional switches. Fig. 7(a)(b)(c) indicate the training errors which the comparisons between ANN outputs and Spice simulations are illustrated in Fig. 7(d)(e)(f). The testing loss for the 3 switches are 8.88×10^{-6}, 3.63×10^{-6} and 2.15×10^{-6} respectively, which proves that small database can train an ANN with high accuracy based on TL. The comparison figures shows the the ANN models provide an accurate estimation of efficiency calculation.



(a) (b) (c)



(d) (e) (f)

Fig. 7: Training loss (a)(b)(c) and efficiency validation (d)(e)(f) for the other switches

The results presented prove that TL is an effective technique to construct an ANN model efficiently. Much less data is needed to be collected for constructing the ANN models. Once all ANN models for the switches list are developed, heuristic algorithms, such as PSO, will be implemented to optimize the parameters.

Parameter optimization

The previous section shows that the ANN models can accurately represent the Spice simulation data. Much less simulation time is needed for Optimization by using the ANN models. A swarm group of 200 is chosen with the random initialization. 4 ANN models were used to be optimized respectively, the optimal values curves are shown in Fig. 8. The current ripple and voltage ripple limits are set as 10% and 1% respectively.

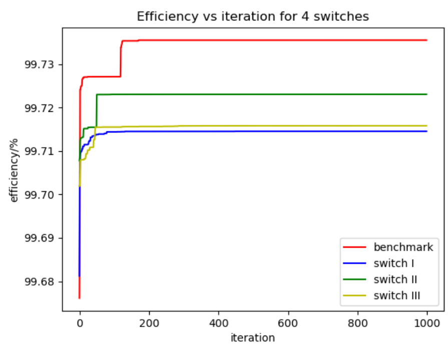


Fig. 8: Optimal efficiency via epoch in PSO algorithm

The best parameters selections are frequency of 30.367 $[kHz]$, inductance of 1.43 $[mH]$, and capacitance of 5.05 μF, with the benchmark switch, in which case the efficiency can reach 97.37% within the ripple constraints.

Conclusion

This paper proposed an AI-based DC-DC buck converter efficiency modelling and parameters optimization method. Spice simulations are used to generate data, which are collected to train an ANN model. TL is implemented to accelerate the process of modelling the power losses of different switches. Finally, the parameters are optimized to maximize the efficiency while considering the ripple constraints. Results indicate that TL is an effective method for training multiple ANNs in similar domains.

References

[1] Martinez W, Yamamoto M, Imaoka J, Velandia F and Cortes C. A.: Efficiency optimization of a two-phase interleaved boost DC-DC converter for electric vehicle applications, IPEMC-ECCE Asia 2016, pp. 2474-2480

[2] Benzaquen J, He J and Mirafzal B.: Toward more electric powertrains in aircraft: Technical challenges and advancements, CES Transactions on Electrical Machines and Systems vol. 5, no. 3, pp. 177-193, 2021

[3] Aguirre M and Yazdani A.: A Single-Phase dc-ac Dual-Active-Bridge Based Resonant Converter For Grid-Connected Photovoltaic (PV) Applications, EPE 2019, pp. 1-10

[4] Li H, Lee S. R, Luo M, Sullivan C. R, Chen Y and Chen M.: MagNet: A Machine Learning Framework for Magnetic Core Loss Modeling, 2020 IEEE 21st Workshop on Control and Modeling for Power Electronics (COMPEL), pp. 1-8

[5] Pan S. J and Yang Q,: A Survey on Transfer Learning, IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359

[6] Dogariu E, Li H, Serrano López D, Wang S, Luo M and Chen M.: Transfer Learning Methods for Magnetic Core Loss Modeling, 2021 IEEE Workshop on Control and Modeling of Power Electronics (COMPEL)

[7] Guillod T, Papamanolis P and Kolar J. W.: Artificial Neural Network (ANN) Based Fast and Accurate Inductor Modeling and Design, IEEE Open Journal of Power Electronics, vol. 1, pp. 284-299, 2020

[8] Li X, Zhang X, Lin F and Blaabjerg F.: Artificial-Intelligence-Based Design (AI-D) for Circuit Parameters of Power Converters, IEEE Transactions on Industrial Electronics

[9] Vishay Siliconix, Appl. Note AN608A, pp.1-6. Available: https://www.vishay.com/docs/73217/an608a.pdf.

[10] del Valle Y, Venayagamoorthy G. K, Mohagheghi S, Hernandez J. -C and Harley R. G.: Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems, IEEE Transactions on Evolutionary Computation, vol. 12, no. 2, pp. 171-195, 2008