RESEARCH LETTER Open Access

Streamlining hyperparameter optimization for radiation emulator training with automated Sherpa

Soonyoung Roh[1], Park Sa Kim[2] and Hwan-Jin Song[3,4*]

Abstract

This study aimed to identify the optimal configuration for neural network (NN) emulators in numerical weather prediction, minimizing trial and error by comparing emulator performance across multiple hidden layers (1–5 layers), as automatically defined by the Sherpa library. Our findings revealed that Sherpa-applied emulators consistently demonstrated good results and stable performance with low errors in numerical simulations. The optimal configurations were observed with one and two hidden layers, improving results when two hidden layers were employed. The Sherpa-defined average neurons per hidden layer ranged between 153 and 440, resulting in a speedup relative to the CNT of 7–12 times. These results provide valuable insights for developing radiative physical NN emulators. Utilizing automatically determined hyperparameters can effectively reduce trial-and-error processes while maintaining stable outcomes. However, further experimentation is needed to establish the most suitable hyperparameter values that balance both speed and accuracy, as this study did not identify optimized values for all hyperparameters.

Highlights

- The study aimed to enhance the efficiency of neural network emulators for numerical weather prediction by reducing trial and error through the use of Sherpa-defined neurons across multiple hidden layers.
- Optimal configurations with one and two hidden layers were identified, leading to a speed increase of 7–12 times and maintaining stable performance with minimal errors.
- While automatically determined hyperparameters have shown promise in decreasing trial and error, additional research is needed to achieve the optimal balance between speed and accuracy in weather prediction.

Keywords Sherpa library, Hyperparameter optimization, Neural-network emulators, Numerical weather prediction, RRTMG-K radiation

*Correspondence:
Hwan-Jin Song
hwanjinsong@gmail.com
Full list of author information is available at the end of the article

Introduction

Abnormal weather patterns, encompassing extreme heatwaves, severe storms, and prolonged droughts, have emerged as a significant global concern. Improving the accuracy of predictions for sudden local heavy rainfalls caused by the tropicalization of the Korean peninsula, the implementation of high-resolution and computationally intensive numerical weather prediction (NWP) models is necessary. Machine learning (ML) technology has become popular as an alternative to providing high-level rapid service in huge NWP models. However, applying ML to physics parameterization, where direct weather prediction is calculated, presents more challenges compared to its use in data assimilation and pre-/post-processing. This is because the interactions between complex equations of physics parameterization throughout the model require sophisticated handling to suppress the prediction error over time. Radiation parameterization, which is a significant factor controlling the energy circulation of the Earth, consisting of scattering, penetration, and reflection of radiant energy is notably complex. As a result, radiation puts a significant burden on driving high-resolution models. Traditional approaches to alleviate this burden have involved intermittent driving of the radiation parameterization. Recently, organizations including the European Center for Medium-Range Weather Forecasts (ECMWF), National Oceanic and Atmospheric Administration (NOAA), and various national meteorological administrations, have been advocating for the use of state-of-the-art ML technology (Chevallier et al. 1998, 2000; Krasnopolsky et al. 2005, 2008, 2010, 2012; Lagerquist et al. 2021; Pal et al. 2019; Roh and Song 2020; Song and Roh 2021; Song et al. 2021; Song and Kim 2022; Song et al. 2022; Ukkonen et al. 2020; Veerman et al. 2021) to facilitate radiation parameterization.

Previous studies (Chevallier et al. 1998, 2000) have investigated the use of emulators for radiation, specifically for LW (longwave radiation) and SW (shortwave radiation), under clear and cloudy sky conditions. The ECMWF devised an advanced LW emulator, termed the control run (CNT), exhibiting a tenfold acceleration compared to the initial radiation scheme. This emulator employs a variable neuron count with a single hidden layer neural network (SHLNN), and utilizes a hyperbolic tangent (tanh) as the activation function (AF). The root-mean-square error (RMSE) between the emulator and CNT was less than 1.7 W m^{-2}. Similarly, the Climate Forecasting System model of the National Centers for Environmental Prediction (NCEP) developed a LW emulator that was 16–20 times faster and had 556 inputs, 69 outputs, and 50–200 neurons SHLNN. The SW emulator was 60 times faster on average, with 562 inputs, 73 outputs, and 50–200 neurons. Krasnopolsky et al. (2012) replaced the LW/SW radiations of the Rapid Radiative Transfer Model for General Circulation Models (RRTMG) with an NN emulator, respectively, resulting in a 20- to 100-fold increase in speed compared to the original radiation parameterization. Bellochiski et al. (2011) explored various methods for LW emulators, including approximate nearest neighborhood, classification, and regression tree (CART) and random forest methods, all of which exhibited higher RMSEs than those using a SHLNN configuration with 80 neurons. The Department of Energy Super-Parameterized Energy Exascale Earth System Model developed an NN-based emulator for LW and SW based on RRTMG-P. This NN consists of 32 neurons across three hidden layers, utilizes a sigmoid activation function, and achieves a speedup factor of 10 with a 90–95% accuracy compared to the CNT. Liu et al. (2020) reported that emulators based on a convolutional neural network (CNN) reduced the RMSE of the LW cooling rate by 41–51% compared with those using a deep neural network (DNN) with three hidden layers, although the CNN was approximately 100 times slower than the DNN. Additionally, AI-based methods for radiative transfer parameterization have also been developed, including gas optical characteristic parameterization (Ukkonen et al. 2020; Veerman et al. 2021) and SW radiative transfer parameterization methods (Lagerquist et al. 2021) for RRTMG-P. However, these emulators have not yet tested online nor have they been fully integrated with operational numerical weather models. Veerman et al. (2021) investigated the performance of emulators that use a combination of neurons and hidden layers. They found that architectures with a high number of neurons in deep layers had small RMSEs but were computationally intensive. In the best performance, the emulator accurately predicted optical characteristics with an average error of less than 0.5 W m^{-2}.

Recent studies by Roh and Song (2020), Song and Roh (2021, 2023) and Song et al. (2021, 2022), have significantly advanced the use of neural network (NN) emulators for radiation parameterization in numerical weather prediction (NWP), focusing on the RRTMG-K radiation scheme, detailed by Baek (2017). Beginning with idealized cases (Roh and Song (2020) and extending to real-world applications, particularly precipitation events in the Korean Peninsula, these works have demonstrated the effectiveness of NN emulators at a high resolution and universal application (Song and Roh 2023). Leveraging the Stochastic Weight Averaging technique, Song et al. (2022) significantly advanced the field by accelerating radiative physics parameterization speeds by a factor of 60, which led to a substantial reduction in computing times by 84–87%. This enhancement was further complemented by improvements in the performance of the emulator, showcasing a 12.30% increase in longwave flux, a 7.16% increase in shortwave flux, and a 3.23% increase in skin temperature, alongside a slight 0.56% decrease in 2-m temperature. These advancements underscore the superior efficacy of the emulator and forecast precision improvements of 18.2–26.9% over the SHLNN model from Song et al. (2021), affirming the operational effectiveness of the emulators.

These advancements underscore the significance of techniques like SWA in hyperparameter optimization, emphasizing the need for diverse approaches to refine hyperparameter settings. This research, by exploring various techniques and balancing computational efficiency with predictive accuracy, not only achieves breakthroughs in atmospheric modeling, but also heralds a promising future for NNs in environmental simulations. The above research demonstrates that optimizing the performance of ML models hinges on emulator design and meticulous parameter construction. Consequently, achieving peak effectiveness is imperative through hyperparameters, such as neuron count, layer quantity, and epochs, as well as input/output training data set regularization, batch size, optimal options, AF, and DNN-based ML learning rate (LR). Most studies have empirically determined optimized hyperparameters during training and validation, leading to numerous cases that must be evaluated to select the optimal choice. Recently, several packages offering various optimization methods for hyperparameters, such as the Sherpa software, have become available. Sherpa can streamline the trial-and-error process using powerful and versatile algorithms (Hertel et al. 2020). For instance, Kim and Song (2022) used the Sherpa-automated optimizer to obtain the optimal LR for ML training, resulting in substantially better performance compared to manually controlled LRs. Optimization technology of Sherpa also significantly reduced the time required for hyperparameter optimization.

These advancements emphasize the importance of hyperparameter optimization techniques (Akiba et al. 2019; Gustafson 2018; Liaw et al. 2018) such as SWA, highlighting the necessity for a variety of methods to fine-tune hyperparameter settings for ML models. This research explores multiple techniques to achieve a balance between computational efficiency and predictive accuracy, leading to significant breakthroughs in atmospheric modeling and forecasting a promising future for the use of NNs in environmental simulations. The optimization of ML model performance is shown to be crucially dependent on emulator design and meticulous parameter construction, including critical hyperparameters such as neuron count, layer quantity, epochs, input/output training dataset regularization, batch size, optimal options, AF, and LR. While most studies have relied on empirical methods to determine optimized hyperparameters during training and validation, the advent of optimization packages like Sherpa software offers a new avenue for streamlining the trial-and-error process with powerful and versatile algorithms (Hertel et al. 2020). For example, the utilization of Sherpa-automated optimizer by Kim and Song (2022) to find the optimal LR for ML training significantly improved performance compared to manually adjusted LRs and reduced the time required for hyperparameter optimization.

Crucially, the development of emulators is significantly influenced by the number of hidden layers and the overall neuron count. Generally, an increase in the number of hidden layers and neurons is associated with enhanced NN training model performance for complex input–output structures. However, this improvement is not universal and applies selectively based on specific circumstances. Given that the input and output structures are predetermined through optimization processes and are immutable, the neuron count emerges as a pivotal factor representing emulator speed, occasionally exhibiting an inverse relationship with performance metrics. This underscores the necessity to predict the optimal number of hidden layers and neurons tailored to the specific problem, a task that is challenging and time-intensive when solely relying on empirical evidence. The use of predetermined values for epochs, LR, optimizers, and data normalization, derived from previous experience, aims to concentrate efforts on the automatic optimization of neuron numbers and hidden layers. This approach enables speed enhancements through computational complexity analysis. Extending the findings of Kim and Song (2022), this study seeks to evaluate the performance of automatically optimized neurons across a spectrum of NN performance, from minimum to maximum. Utilizing data from 2009 to 2020 provided by Song et al. (2022), the research will employ the SWA technique with Sherpa software to predict the optimal configurations of hidden layers and corresponding neurons, integrating these insights to push the boundaries of emulator efficiency and NN performance in environmental modeling.

Data and methods

Sherpa model

Sherpa is a library focused on automating the hyperparameter optimization process for NNs, aiming to maximize model performance. It seeks to optimize crucial variables in network structure and the learning process, such as the number of neurons per layer, by defining a parameter search space and utilizing a genetic algorithm to find the optimal values. In our experiments, we adjusted the neuron parameter within a range of 5–1098

(Table 1), with the optimization limited to a maximum of five attempts. The basic definitions used in Sherpa experiments encompass various optimization and model configuration components. The maximum number of epochs, which determines the repetition count for the learning process of the model, was limited to 3000 (clear sky) and 2200 (cloudy sky) proposed by Song et al. (2022), focusing our experiment on the dependency on neuron numbers. The learning rate is set at a default of 0.5 (Kim and Song 2022), with the SGD (Stochastic Gradient Descent) optimizer applying a momentum of 0.9, enhancing adjustability during the learning process. Gradient clipping is also employed to prevent gradient explosion issues associated with gradient descent methods. The structure of the model employs a multi-layer perceptron (MLP), with the number of neurons in the hidden layers optimized through Sherpa to maximize model performance. In this case, we are dealing with fully connected feed-forward NNs that feature 1–5 hidden layers, with the same number of neurons in each hidden layer. A batch size of 500 (Kim and Song 2022) increases data processing efficiency. The models employed a tanh AF. And CosineAnnealingLR scheduler was used for learning rate adjustment. The loss function primarily used is MSE (mean squared error). Additional settings include batch normalization, minimum variance settings, Early Stopping conditions, and checkpoint saving, optimizing the model learning and validation process, preventing overfitting, and enhancing the generalization capability of model.

In this case, the application of SWA improves learning stability. SWA models, introduced by Izmailov et al. in 2018, were employed to enhance the generalization of NN training. SWA is a machine learning technique that improves the generalization performance of NN training by averaging the network weights at various points during SGD. Unlike traditional ensemble methods, SWA is computationally efficient and results in broader, flatter solutions that enhance generalization, as opposed to SGD, which can converge to sharp minima. Izmailov et al. (2018) demonstrated the superior performance of SWA in benchmark tests compared to SGD. In the context of emulators for General Circulation Models and NWP, SWA shows promise in addressing generalization challenges caused by accumulated errors during long-term integration. Generalization remains a significant concern when developing universal emulators, especially in cases where an infinite amount of training data is not available. The SWA mode was applied during the final 25% of epochs, in contrast to the initial 75% utilizing conventional SGD.

The use of Sherpa significantly reduces the manual labor and time required for tuning of the model by swiftly identifying the most efficient configuration among various options. This optimization tool automates the process of enhancing the ability of the model to learn from and predict data, thereby improving both accuracy and execution speed. Applying a scientific experiment methodology, it adjusts the impact of specific hyperparameters to achieve optimal performance. The goal is to find

Table 1 Number of neurons (N) derived from NC

		Number of hidden layers (H)				
		1	**2**	**3**	**4**	**5**
Speedup (fold)	5	1098 (264,660)	392 (248,570)	295 (245,777)	248 (245,066)	218 (243,548)
	10	549 (132,351)	253 (125,277)	196 (124,502)	166 (123,214)	147 (122,493)
	20	274 (66,076)	158 (63,242)	126 (62,412)	109 (62,281)	98 (62,468)
	50	110 (26,552)	80 (25,802)	68 (25,814)	60 (25,482)	54 (24,936)
	100	55 (13,297)	45 (12,957)	40 (12,962)	36 (12,714)	34 (12,996)
	200	27 (6549)	24 (6426)	22 (6356)	21 (6489)	20 (6542)
	500	11 (2693)	10 (2562)	10 (2672)	10 (2782)	9 (2571)
	1000	5 (1247)	5 (1277)	5 (1307)	5 (1337)	5 (1367)

Parentheses show the NC of the given mean input (I = 175), output (O = 42), and the number of hidden layers (H) the best hyperparameter configuration that maximizes model performance while minimizing computational costs. Sherpa enables learning and improvement through trial and error, offering a considerable advantage by facilitating rapid experimentation cycles to efficiently identify the best model configuration.

Training and validation sets

In the real-case study, a horizontal domain featuring a 234×282 grid layout for east–west and north–south directions was employed, covering the Korean peninsula. This setup mirrors the configuration of the Korea Local Analysis and Prediction System (KLAPS), an operational numerical weather prediction (NWP) model utilized by the Korea Meteorological Administration (KMA). KLAPS is designed for short-range weather forecasting, with a primary focus on the rapid analysis of changing weather phenomena over brief periods, as documented by Shin et al. (2022). The system operates with a 5-km spatial resolution and is composed of 39 vertical layers, extending up to 50 hPa. Note that the dynamics and physical processes of KLAPS are grounded in the Weather Research and Forecasting (WRF) model (Skamarock et al. 2019). The radiation emulator for this framework targets the RRTMG-K radiation scheme (Baek 2017), capable of calculating vertical heating rates, LW fluxes across 16 bands using 256-g points, and SW fluxes across 14 bands using 224-g points. The simulation incorporated the WRF double moment 7-Class microphysics scheme (Bae et al. 2019), the KIAPS Simplified Arakawa–Schubert cumulus (Kwon and Hong 2017), the Shin and Hong planetary boundary layer (Shin and Hong 2015), the revised MM5 Monin–Obukhov surface layer (Jiménez et al. 2012), and the Unified Noah land surface model (Tewari et al. 2004). Initialization for the real case framework was done using data from the European Center for Medium-Range Weather Forecasts Reanalysis v5 (ERA5) (Hersbach et al. 2020) with a 0.25° grid and 3-h intervals.

In this study, we adopted the CNT established by Song et al. (2022) to develop the RRTMG-K emulator operating within the KLAPS environment using WRF, covering the period from 2009 to 2020. Our focus was on July, which is known as a period when severe weather events frequently occur on the Korean peninsula, for the development of the emulator through the Sherpa experiment. The data used for this study comprised July of each year from 2009 to 2019 as the training set. We targeted July for the period from 2009 to 2019, selecting 2 days of maximum precipitation and obtaining 2 days without precipitation randomly to use training data for a total of 44 days. The independent validation sets were composed of days corresponding to the third and fourth maximum precipitation events in July during the period from 2009 to 2019, along with 22 non-precipitating days that were not used in the training sets. This segmentation was strategically chosen to ensure a comprehensive training and validation process for the emulator. The rationale behind using a large training dataset is to adequately cover the complexity of weather situations and address the uncertainties in meteorological variables caused by factors such as complex weather patterns and global warming. Our objective was to create an emulator capable of replacing the RRTMG-K radiation scheme in the WRF model, enabling the calculation of radiation flux and heating rates solely through NN inference. To assess the emulator performance in an operational setting, an independent verification targeted July 2020 for an online test. The results of this online test, which demonstrate the accurate prediction of radiation fluxes and heating rates by the emulator without relying on the traditional RRTMG-K radiation scheme, are detailed in chapter 3.

Traditionally, radiative transfer parameterization involves conducting one-dimensional numerical calculations of the radiative heating rate at specific grid points. Input variables encompassed vertical profiles of pressure, temperature, water vapor, ozone, and cloud fraction, skin temperature (LW) and surface emissivity (LW), insolation (SW), and surface albedo (SW). Output variables included all-sky heating rate profiles, upward fluxes at both the top and bottom of the atmosphere, and downward flux at the bottom. In this study, LW/SW fluxes were calculated as the average of the three fluxes at both the upward fluxes and downward flux. Individual fluxes at the top or bottom were also analyzed statistically. The datasets were divided into eight groups based on radiation types (LW and SW), atmospheric conditions (clear and cloudy sky), and geographic scenarios (land and ocean). Surface data (land or ocean) and cloud fraction data (clear or cloudy sky) provided horizontal information, along with time data for solar angles (applied to LW or SW) for specific months. Solar zenith angle determined the application of LW and SW radiation, with LW active during nighttime when the solar zenith angle is negative and both LW and SW active during daytime when the solar zenith angle is positive. Each category contains approximately 3 million data points.

NN complexity

The NC of a DNN can be defined as:

$$\begin{aligned} \mathrm{NC} &= \mathrm{I} \times \mathrm{N} + \mathrm{N} + (\mathrm{H} - 1) \\ &\quad \times (\mathrm{N} \times \mathrm{N} + \mathrm{N}) + \mathrm{N} \times \mathrm{O} + \mathrm{O}, \end{aligned} \tag{1}$$

where I and O denote the input and output variables, respectively, N represents the number of neurons per hidden layer, and H corresponds to the number of hidden layers. This complexity enables the calculation of the speedup based on N and H, considering the dimensions

(I and O) of the provided training data. Notably, the NWP model supplies cloud fraction data across 39 vertical layers; however, observations reveal that the upper 4 layers (layers 36 to 39) are typically devoid of clouds in July. To enhance performance, we refined our dataset to focus on the 35 lower layers for modeling conditions of a cloudy sky, effectively reducing NC and, consequently, accelerating the NN. In our model for a cloudy sky, I is set to 193, reflecting the inclusion of data from these 35 layers. For conditions of a clear sky, where the vertical cloud fraction values at the numerical grid points equal 0, we omitted the cloud fraction data for all 39 layers from the input, resulting in 158 variables. The output O, which represents 8 categories in the training, is fixed at 42. Based on the I/O provided by Song et al. 2021 and Song et al. 2022, we adopted the result that using 90 neurons led to a 60-fold speed improvement in the emulator, as measured by NC. Therefore, future calculations of speedup were performed based on this benchmark.

Performance of Sherpa in the NN-based emulator

Speedup

Table 1 details the structure of NNs, specifying the number of neurons across varying hidden layers with a set mean input of 175, indicative of average clear and cloudy sky conditions, and a fixed output of 42. The exploration spans one to five hidden layers (H), with neuron counts adjusted for speed enhancements ranging from 5- to 1000-fold to optimize experimental efficiency. For the fivefold speedup, 1098 neurons are used, while at 1000-fold, the count is reduced to just 5 neurons. Ultimately, this establishes a fundamental framework for utilizing Sherpa to search for an optimal neuron count within the range of 5 to 1098. The NC is calculated to quantify of NN structural complexity, facilitating comparison in terms of size and architecture. Higher NC values correlate with an increased number of neurons and connections, suggesting a potential to learn more complex patterns. However, this complexity can also predispose networks to overfitting and elevate computational demands (Bellochiski et al. 2011, Belochitski and Krasnopolsky 2021, Song et al. 2022). Certainly, the table shows the reduction in neuron quantity with accelerated computational speeds, implying a preference for simpler network structures for faster processing. Additionally, in this framework, an increase in hidden layers corresponds with a decrease in neurons per layer, indicating a balance between network depth and complexity management. In summary, the table demonstrates how increases in speed and hidden layer count influence neuron numbers and NC, serving as a vital reference for NN design and optimization.

Table 2 illustrates the outcomes derived using the Sherpa algorithm, encompassing results for neuron, speedup, and reduction rates across 1 to 5 hidden layers. The data are segregated based on LW and SW radiation, clear and cloudy skies, and land and ocean scenarios. The metrics for speedup and reduction are computed by averaging the neuron counts deduced by Sherpa for the respective hidden layers. These neurons are obtained through five iterations using genetic algorithm of Sherpa, demonstrating the algorithm efficiency in optimizing NN configurations under various environmental conditions and hidden layer quantities. The speedup and reduction metrics serve as crucial elements for improving the NN performance and efficiency, offering essential information for radiative physics calculations and applications across different meteorological and geographical environment. Through experimentation, we explored the

Table 2 Neurons, speedup and reduction across 1 to 5 hidden layers as derived from Sherpa

Atmospheric condition	Radiation type	Geographic type	Number of hidden layers (H)				
			1	2	3	4	5
Clear sky	LW	Land	436	379	231	219	166
		Ocean	966	119	249	184	168
	SW	Land	505	379	274	242	144
		Ocean	130	79	138	156	148
Cloudy sky	LW	Land	471	370	263	237	123
		Ocean	384	275	253	187	138
	SW	Land	443	217	190	234	206
		Ocean	188	304	284	164	130
Sherpa-derived average neurons			440.4	265.3	235.3	202.9	152.9
Speedup (fold)			12.4	9.3	7.4	7.1	9.4
Reduction (%)			18.1	13.5	10.7	10.3	13.6

It includes distinctions for LW and SW radiation, clear and cloudy sky, and land and ocean scenarios. The speedup and reduction are based on Sherpa-derived average neurons

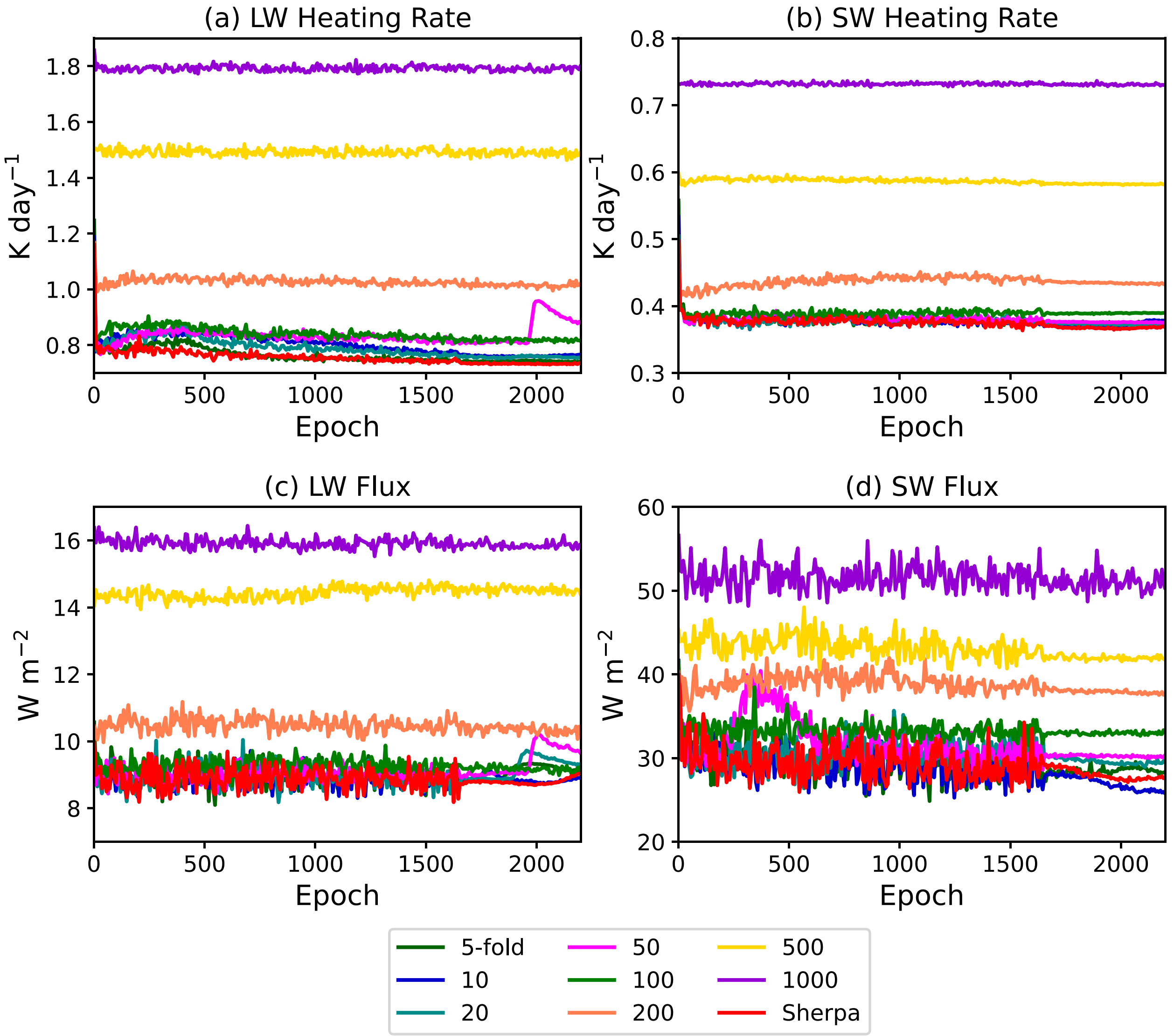

Fig. 1 Learning curves displaying the RMSEs from emulators with various speedups (5, 10, 20, 50, 100, 200, 500, and 1000) and Sherpa-applied emulator with SHLNN. The maximum epoch was set to 2200. Each line represents the average of four categories (land, ocean, clear, and cloudy sky)

correlation between optimized neurons and categories. A trend was observed where the required neuron decreases with an increase in the number of hidden layers, suggesting that DNNs do not necessarily require more neurons, highlighting the importance of finding a balance between efficiency and complexity. Depending on the type of radiation, LW generally necessitates more neurons than SW, likely due to closer of LW radiation association with the complex thermal characteristics of the Earth surface. Conversely, in SW, NNs can effectively learn with fewer neurons, primarily because solar radiation calculations are more direct and based on specific values. Under different atmospheric conditions, clear skies enable a clearer learning of meteorological variables by the NN due to the absence of clouds, necessitating the network to finely detect and predict subtle changes in variables like surface temperature, humidity, and albedo. In cloudy conditions, the presence of clouds directly affects the transmission and distribution of radiative energy. Geographically, terrestrial areas, due to their rapid heating and cooling, can generate complex weather patterns, implying a need for more neurons in the NN. In contrast, the ocean shows more uniform and steady temperature changes, allowing the NN to achieve sufficient predictive performance with fewer neurons. This analysis provides important insights into the complexities of LW and SW radiation, clear and cloudy skies, and land and ocean conditions. It confirms the significant impact of each condition physical

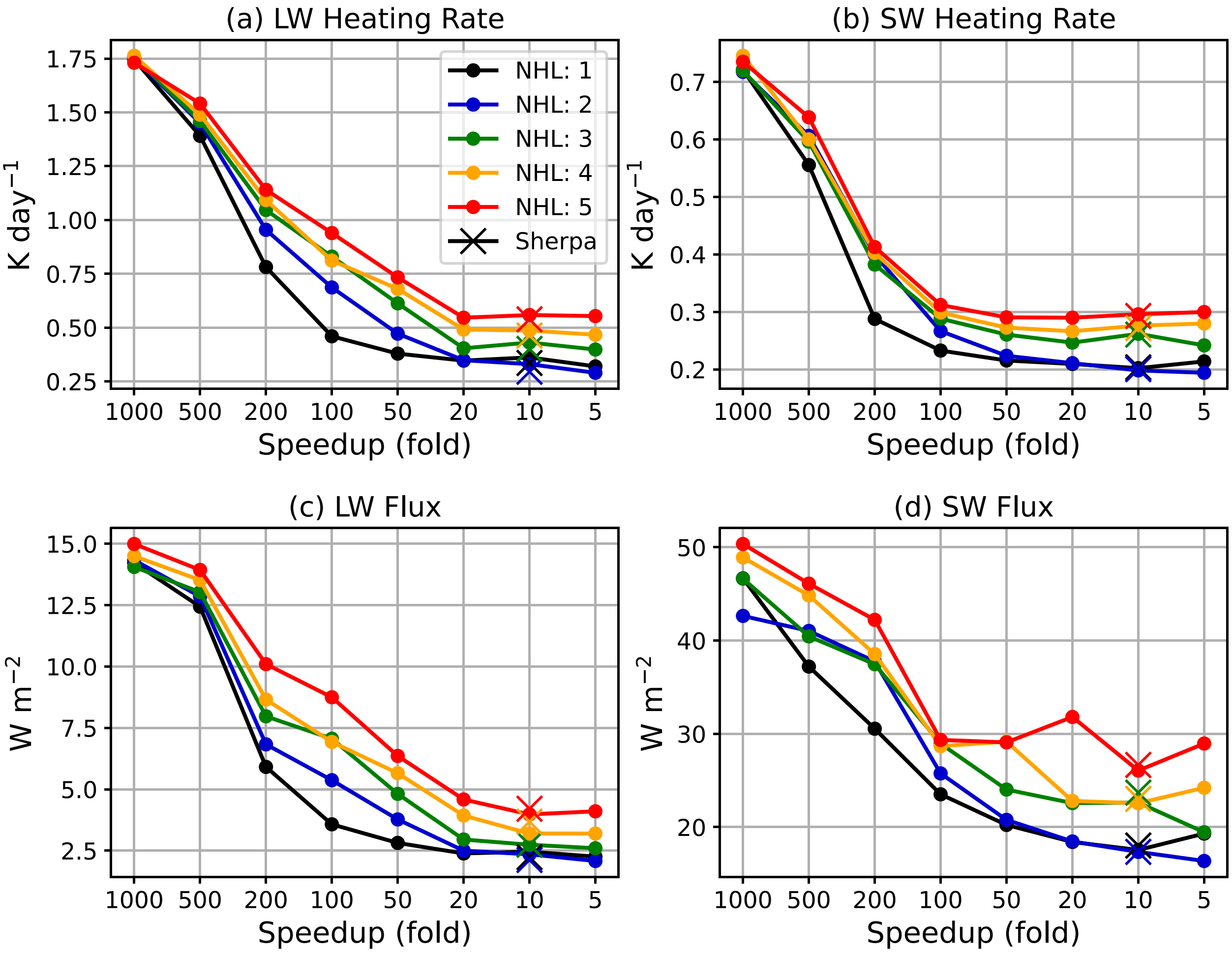

Fig. 2 RMSEs of LW/SW heating rates and fluxes from emulators with varying speedups (•) and Sherpa-applied emulator (×). Different colors denote number of hidden layer (NHL), ranging from 1 to 5

characteristics and the interaction with meteorological variables on the model structure and computational capabilities. According to the study by Song et al. (2022), the radiation emulator achieved a speed enhancement of 60-fold compared to the CNT, reducing the total computational time of NWP by approximately 87%. Using linear interpolation, it was determined that at a maximum speed enhancement of 1000 times, the computational time is reduced by 1450% compared to the CNT, and at a minimum speed enhancement of 5 times, it is reduced by 7.3%. The NN emulators, utilizing neuron counts automatically determined by Sherpa across 1 to 5 hidden layers, showed a 7–12 times speed enhancement (average 9 times), consequently reducing the total computational time of NWP by 10–18% (average 13.2%).

Accuracy

Figure 1 illustrates the learning curves with RMSEs for the SHLNN, detailed in Table 2. The training was stopped at 3000 epochs for clear skies and 2200 epochs for cloudy sky. Here, the comparison includes average results up to 2200 epochs. In the figure, various colors indicate results from different speedup conditions, while solid red line denotes the Sherpa result (12.5-fold speedup as detailed in Table 2). Here, (a) and (b) account for the averaged heating rates summed across vertical layers. Figure 1c and d represent the average of 3 fluxes: upward fluxes at the top and bottom of the atmosphere, and downward flux at the bottom for LW and SW, respectively. Each learning curve is an average of four categories (land, ocean, clear, and cloudy sky) and corresponds to speedups of 5, 10, 20, 50, 100, 200, 500, and 1000-fold, as well as Sherpa experiment with SHLNN. Generally, RMSEs slightly decreased with an increase in epochs. In the figure, SGD is applied for epochs less than 1650, whereas constitute approximately 75% of the total 2200 epochs, while the SWA model (Song et al., 2022) exhibits quasi-stable RMSEs for the remaining 25% of epochs. For heating

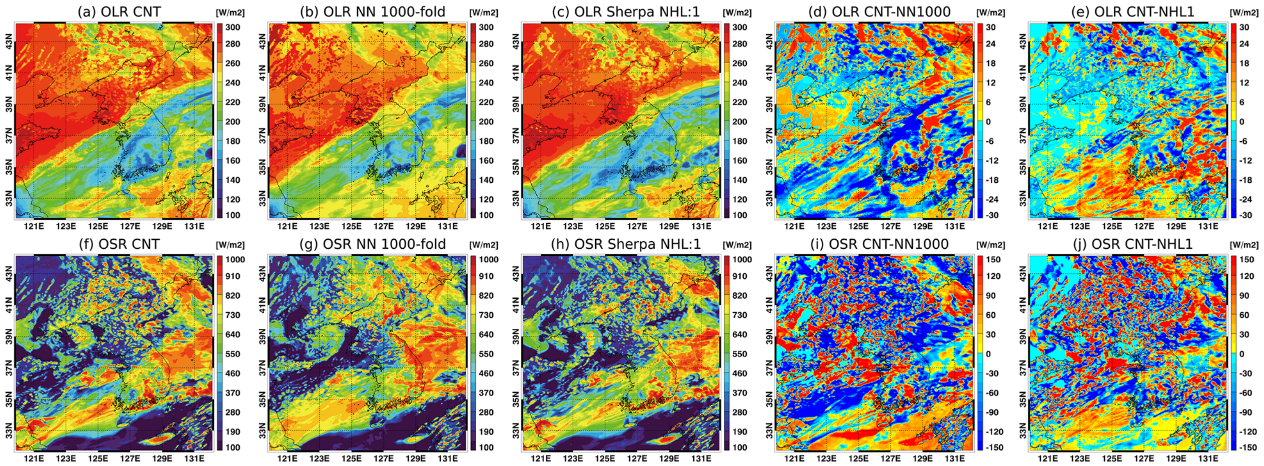

Fig. 3 Spatial distribution of fluxes. The top row display OLR, while the bottom row show OSR, respectively. It represents the CNT (**a, f**), the emulator with a NN 1000-fold speedup in SHLNN (**b, g**), the emulator resulting in Sherpa (**c, h**), the difference between the CNT and the 1000-fold emulators in SHLNN (**d, i**), and the difference between the CNT and the Sherpa-applied emulator (i.e., SHLNN) emulators (**e, j**) for OLR and OSR, respectively. This image illustrates the spatial distribution for 0300 UTC on 28 July 2020, integrated with the forecast results from 156 to 356 h, initiated at 1500 UTC on 21 July 2020

rate and flux, curves with less than 100-fold speedup had relatively small errors due to high NC. For LW and SW heating rates, the averaged RMSEs with less than a 50-fold speedup at the 2200th epoch was 0.791 day^{-1} and 0.378 K day^{-1}, respectively. The fluxes for LW and SW were 9.238 and 29.225 W m^{-2}, respectively. Compared to the Sherpa result in Table 2, the speedups of 12.28-fold were located between the 10- to 20-fold speedups of Fig. 1. For Sherpa at the 2200th epoch, the RMSEs of LW and SW heating rates and fluxes were 0.736 and 0.369 K day^{-1} and 9.052 and 27.440 W m^{-2}, respectively. Although these values were slightly lower than the LW and SW heating rates of 0.756–0.762 K day^{-1} and 0.373–0.376 K day^{-1}, respectively, they fell within the ranges of 8.904–9.292 W m^{-2} and 25.766–29.324 W m^{-2} for LW and SW fluxes at approximately 10–20-fold speedup at the same epoch, respectively. In summary, the learning curves for different speedup conditions show varying patterns, with Sherpa results generally achieving lower RMSEs compared to other speedup conditions. The relationship between speedup, NC, and model stability influenced the overall performance of the NN emulators.

Figure 2 shows the RMSEs of emulator using neuron determined by the speedup for given NHLs (Table 1) and Sherpa. The circle symbols represent the RMSEs of the heating rate for LW (a) and SW (b), and the average of three fluxes for LW (c) and SW (d), respectively. The figure reveals that, for a given hidden layer, error decreases when there is a minor speed improvement (using a larger number of neurons). For most of the results, barring SW flux, it becomes evident that the deeper the layer is for a specific speed improvement, the larger the error. Among all results, SHLNN demonstrated the best performance. This finding indicates that developing more complex structures in the emulator does not always yield better results, supporting the conclusions of previous studies, such as Song et al., 2022. In other words, this insight can be used in future research as an indicator that helps strike a balance between speed improvement and DNN structure. For SHLNN, the LW heating rate and fluxes had RMSEs of 0.45 K day^{-1} or less and 3.8 W m^{-2} or less for a speed improvement of 100 times or less. The SW showed 0.23 K day^{-1} or less measurements and 24 W m^{-2} or less.

In this experiment, Sherpa were assumed to be comparable to a tenfold improvement. In the experiment with a speedup of 5, the RMSE of Sherpa was smaller than that for the cases with 4–5 hidden layers, and larger than those with 1–3 hidden layers. For the SW heating rate, the RMSE decreased as the speedup was reduced for each hidden layer, becoming nearly constant when the speedup was 50 or below. With a speedup of 5, the case with two hidden layers displayed the smallest RMSE for the SW heating rate. In the Sherpa experiment with one and two hidden layers, the RMSE was almost identical and exhibited the minimum value. The LW flux experienced a decrease in RMSE for 1 to 5 hidden layers as the speedup decreased. When the speedup was less than 10, the RMSE showed minimal reductions across most hidden layers. With a speedup of 5, the RMSE achieved the minimum value in the entire experiment using two hidden layers. Sherpa RMSE was similarly small for one and two hidden layers. However, the SW flux demonstrated

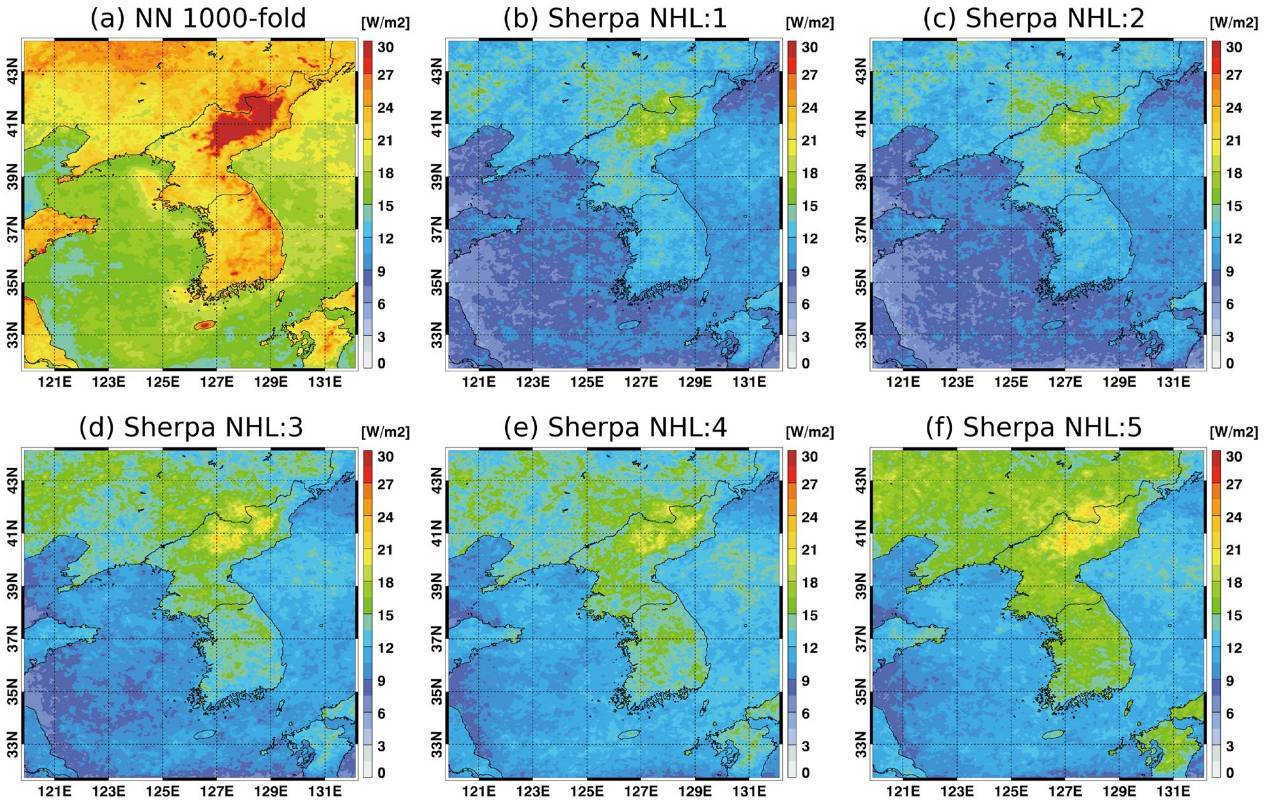

Fig. 4 Spatial distributions of the RMSEs for averaged LW fluxes (upward fluxes at the top and bottom of the atmosphere, and downward flux at the bottom) from emulators trained with a speedup of 1000-fold in SHLNN (**a**) and Sherpa-applied emulator with 1–5 hidden layers (**b**–**f**)

a different trend than the abovementioned variables. The RMSE of SW flux with 1, 4, and 5 hidden layers decreased as the speedup dropped, but the RMSE rose again at a speedup of 5. With five hidden layers, the RMSE of SW flux went down at a speedup of 1000 to 100, increased slightly, and then decreased at a speedup of 20.

Figure 3 exhibits the spatial distributions of OLR fluxes (upper panel) and OSR fluxes (lower panel). The depiction data accrued over 156 h in the WRF amid the extreme weather occurrence marked by the Changma front on the Korean Peninsula. The figures demonstrate the CNT (a and f), the 1000-fold speedup emulator (b and g), and Sherpa-applied emulator (c and h). For comparison, the difference between CNT and experiments is shown in (d and e) for OLR and (e and j) for OSR, respectively. The emulator experiments aimed to discern the maximal difference between the lowest performance (1000-fold speedup) and optimized performance utilizing Sherpa with the SHLNN framework. The OLR depicted the clear sky area with elevated LW radiation flux as a red zone in panel (a–c). Panels (d) and (e) show the difference between CNT and experiments. The difference shown in (e) is smaller than (d). Conversely, OLR fluxes in cloudy sky areas were diminished due to obscured surfaces by dense cloud cover. Comparing the distributions, the Sherpa result was found to be more congruent with (a). The OSR (lower two panels) displayed augmented values where the reflection effect was more pronounced due to cloud-covered surfaces, resulting in reduced OSR flux in areas where the surface absorbed it. Analogous to the LW outcome, the Sherpa-output was more akin to the OSR CNT than the emulator result.

The RMSEs of the spatial distribution, obtained by averaging three of each LW/SW flux and skin temperature, are presented in Figs. 4, 5, 6 to illustrate the extreme difference between the CNT and experiments using various hidden layers. Figures 4a, 5a, and 6a show the RMSE spatial distributions of the CNT versus the emulator designed using neurons with a 1000-fold speedup under the SHLNN. Figures 4b–f, 5b–f, and 6b–f depict the RMSEs obtained by using the number of neuron results in Sherpa that were evaluated by the CNT. The emulator used hidden layers with 1 to 5 layers. All data were collected over 4 weeks from 1500 UTC 21 July 2020 to 1500 UTC 21 August 2020 and predicted for 7 days once. The results of the 4-week prediction were applied to find the RMSEs at each grid point. The LW and SW fluxes were considered as the average of three surface

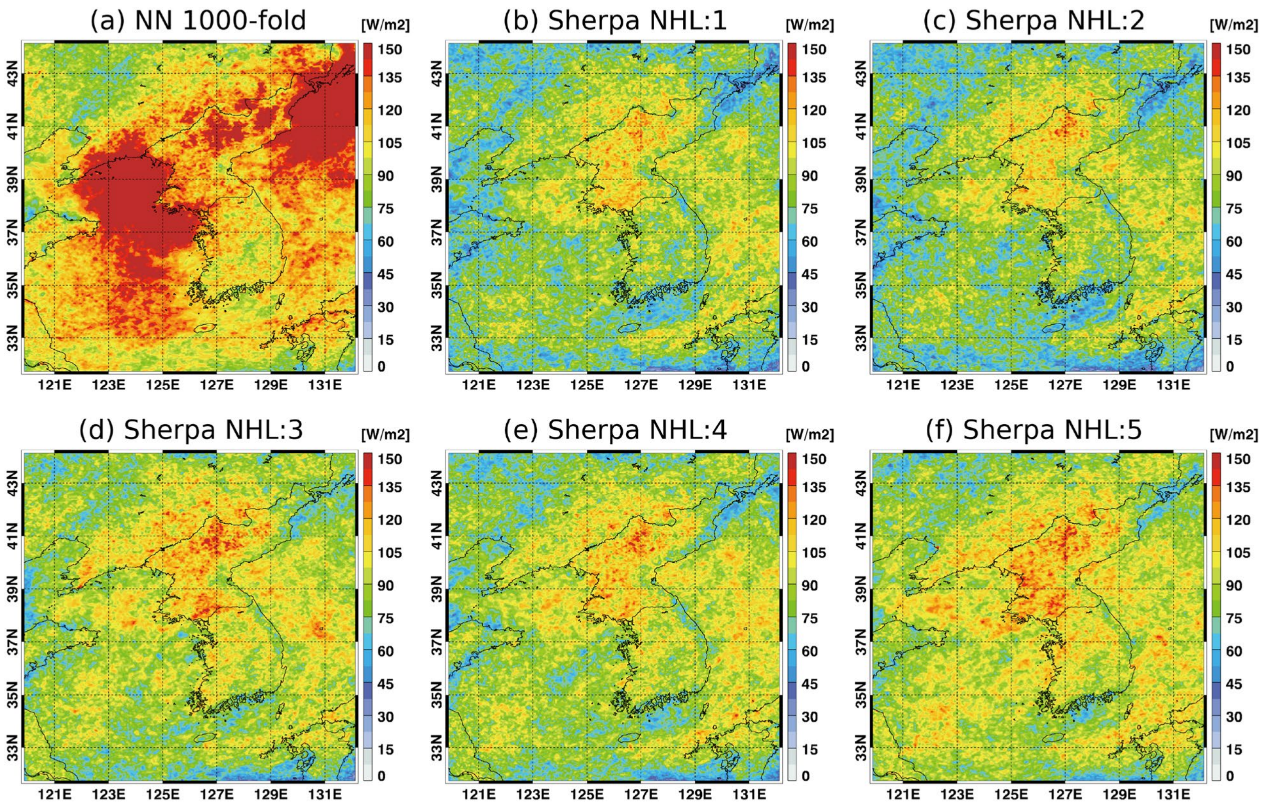

Fig. 5 Spatial distributions of the RMSEs for averaged SW fluxes (upward fluxes at the top and bottom of the atmosphere, and downward flux at the bottom) from emulators trained with a speedup of 1000-fold in SHLNN (**a**) and Sherpa-applied emulator with 1–5 hidden layers (**b**–**f**)

outputs. Overall, the RMSE distribution of NN 1000-fold was larger than that of the Sherpa experiments, as shown in Figures 4b–f, 5b–f, and 6b–f. Furthermore, RMSEs were enhanced towards the deep layers. In detail, NN 1000-fold in Fig. 4 showed large errors in high mountain locations, such as North Korea and Jeju Halla Mountain, because the elevation of the terrain was not considered during the ML training. Moreover, the error for the ocean region was significantly larger, approximately one order of magnitude, than the error in Sherpa experiments. Figure 5a–f shows higher RMSEs with similar patterns for the SW fluxes for the mountainous region in North Korea. However, NN 1000-fold had the largest RMSE (reddening area) not only in the high mountain region (up to 150 W m^{-2}), but also for a broad area of the northern ocean (Fig. 5a). Moreover, overall, the results of NN 1000-fold had higher errors than those of the LW experiment (Fig. 4). For Sherpa experiments, the high mountain location in North Korea still showed large RMSEs distribution, but the emulators with Sherpa experiment well-imitated the ocean regions. This pattern is also well demonstrated by skin temperature in Fig. 6. The skin temperatures for ocean regions were well-emulated with almost zero value. For the land, the northern Korean peninsula is illustrated up to 3 K, and a difference was found between Sherpa experiments showing over 2.5 K.

Figure 7 presents the time series of improvement rates for the RMSE calculated using Sherpa, based on the number of hidden layers, in comparison with the RMSE in SHLNN, which has a speedup of 1000-fold for LW/SW fluxes, skin temperature, and precipitation. The statistics were derived by considering the spatial and time evolutions over a 4-week period. For the RMSE of the LW flux, the improvement rate increased as the number of hidden layers decreased (Fig. 7a). The improvement rate of RMSE with two hidden layers was almost the same or slightly higher than that with one hidden layer. As the 7-day forecast hours progressed, the improvement rate of RMSE exhibited daily fluctuations and subsequently decreased over time. Similarly, the RMSE of the SW flux saw an improvement with a lower number of hidden layers (Fig. 7b). However, the improvement rate of RMSE for the SW flux was slightly higher with two hidden layers compared to one hidden layer. For the 7-day forecast hours, the improvement rate of RMSE diminished over time. It should be noted that only daytime results when SW radiation occurred are displayed.

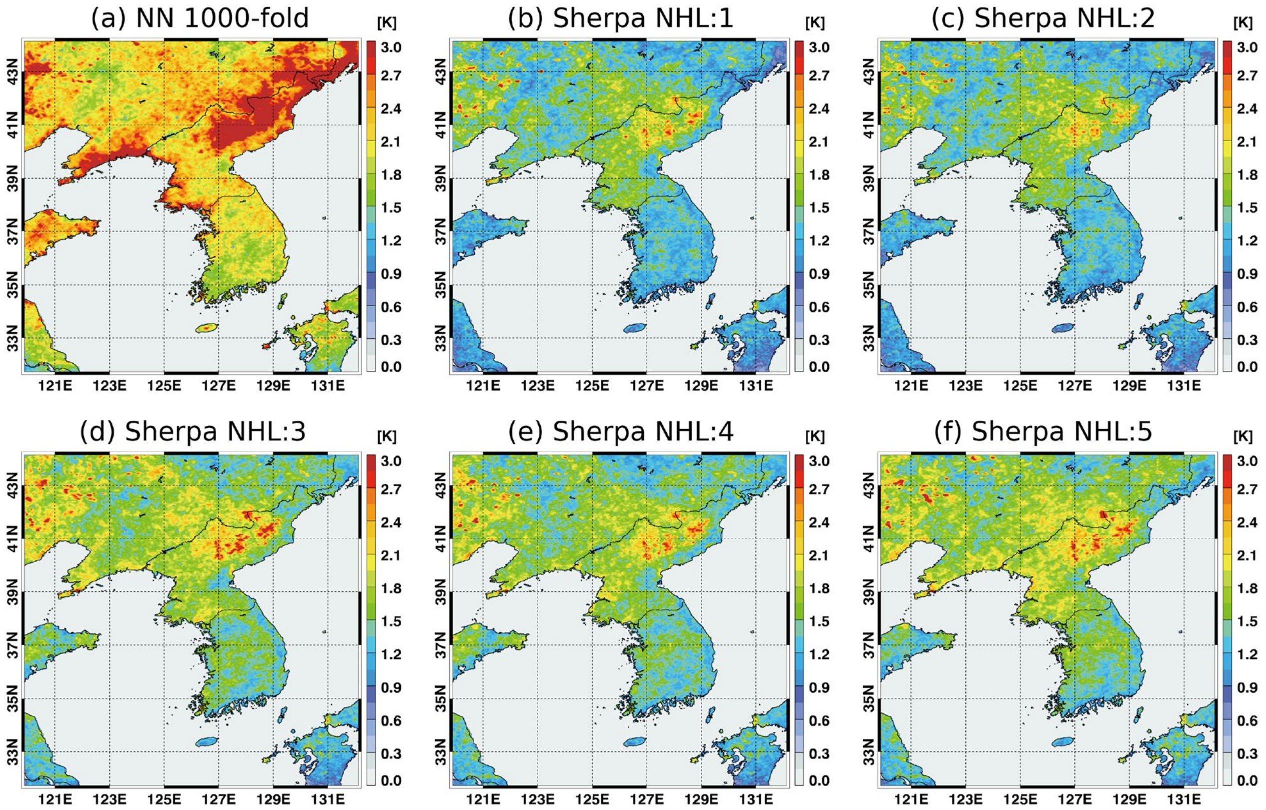

Fig. 6 Spatial distributions of the RMSEs of skin temperature from emulators trained with a speedup of 1000-fold in SHLNN (**a**) and Sherpa-applied emulator with 1 to 5 hidden layers (**b**–**f**)

Regarding the RMSE of skin temperature, the improvement rate increased as the number of hidden layers became smaller, but the difference in the improvement rate of RMSE between the number of hidden layers was not significant (Fig. 7c). Across the 7-day forecast hours, the improvement rate of RMSE exhibited daily fluctuations and decreased over time. Lastly, for precipitation, the improvement rate of RMSE did not display any specific characteristics concerning the hidden layers, and no discernible trend was observed as time progressed.

Summary and conclusions

In this study, we compared the performance of emulators considering neurons obtained from multiple hidden layers (1–5 layers), automatically defined by the Sherpa library, in the context of numerical weather prediction. Our goal was to determine the most efficient and accurate configuration for NN emulators while minimizing trial-and-error processes. Our findings revealed that emulators with neurons determined by Sherpa consistently demonstrated good results and stable performance with low errors in numerical simulations. The optimal configurations were observed with one and two hidden layers, slightly improving results when two hidden layers were employed. The Sherpa-defined average neurons per hidden layer ranged between 153 and 440, resulting in a speedup relative to the CNT of 7–12 times. These results provide valuable insights for developing radiative physical NN emulators. Utilizing automatically determined hyperparameters can effectively reduce trial-and-error processes while maintaining stable outcomes. However, further experimentation is recommended to establish the most suitable hyperparameter values that balance both speed and accuracy, as this study did not identify optimized values for all hyperparameters.

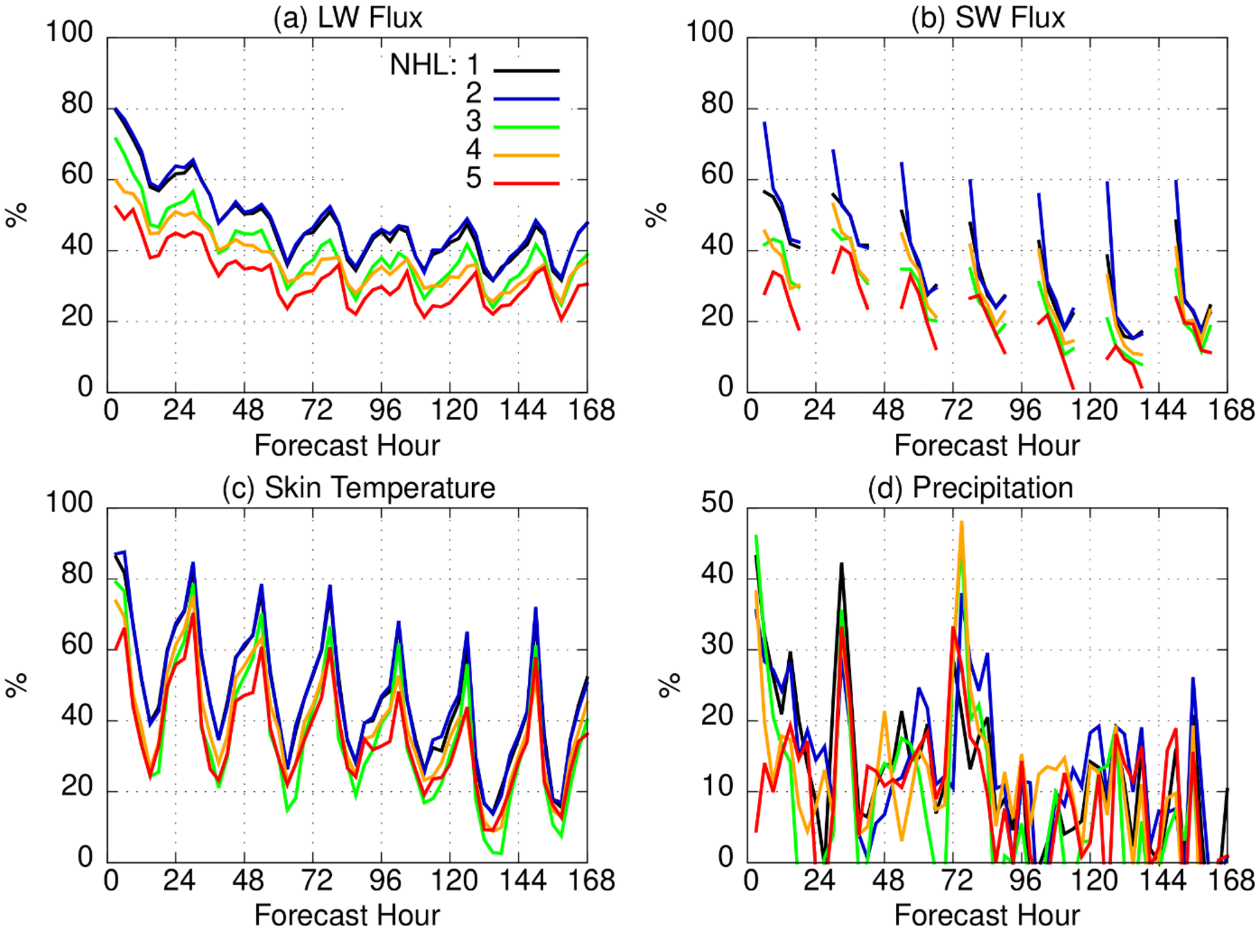

Fig. 7 Time series of the improvement rate of RMSEs. The results of Sherpa with 1–5 hidden layers are compared to the findings obtained with the speedup of 1000-fold. Figures **a** and **b** correspond to LW and SW fluxes, respectively. Skin temperature and precipitation are displayed in **c** and **d**, respectively

Acknowledgements
Dr. S. Roh was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00211994). Dr. P.S. Kim was funded by the Korea Meteorological Administration (KMA) Research and Development Program "Developing AI Technology for Weather Forecasting" under Grant (No. KMA2021-00121). Dr. H.-J. Song gratefully acknowledges financial support by the National Research Foundation of Korea (NRF) Grant funded by the Korea government (MSIT) (No. RS-2023-00210362) and Learning & Academic research institution for Master's·PhD students, and Postdocs (LAMP) Program of the National Research Foundation of Korea (NRF) Grant funded by the Ministry of Education (No. RS-2023-00301914).

Code availability

The neural network radiation scheme, the modeling system used in this study was obtained from Song et al. (2022). The data and codes used in this manuscript are available in the following: https://doi.org/10.5281/zenodo.5638436. Although both systems were based on the WRF model (version 3.9.1.1), they were modified to use the radiation emulator.

Declarations

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author details
[1]Center for Atmospheric REmote Sensing (CARE), Kyungpook National University, Daegu, South Korea. [2]National Institute of Meteorological Sciences, Korea Meteorological Administration, Jeju-do, South Korea. [3]BK21 Weather Extremes Education and Research Team, Department of Atmospheric Sciences, Kyungpook National University, Daegu, South Korea. [4]KNU Institute of Basic Sciences, KNU LAMP Research Center, Kyungpook National University, Daegu, South Korea.

Received: 22 August 2023 Accepted: 28 March 2024
Published online: 13 April 2024

References

Akiba T, Sano S, Yanase T, Ohta T, Koyama M (2019) Optuna: a next-generation hyperparameter optimization framework. Proc. 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining, Anchorage, AK, SIGKDD, 2623–2631, https://doi.org/10.1145/3292500.3330701

Bellochiski A, Binev P, DeVore R, Fox-Rabinovitz MS, Krasnopolsky VM, Lamby P (2011) Tree approximation of the long wave radiation parameterization in the NCAR CAM global climate model. J Comput Appl Math 236:447–460. https://doi.org/10.1016/j.cam.2011.07.013

Chevallier F, Chéruy F, Scott NA, Chédin A (1998) A neural network approach for a fast and accurate computation of a longwave radiative budget. J Appl Meteor 37:1385–1397. https://doi.org/10.1175/1520-0450(1998)037

Chevallier F, Morcrette J-J, Chéruy F, Scott NA (2000) Use of a neural-network-based long-wave radiative-transfer scheme in the ECMWF atmospheric model. Quart J Roy Meteor Soc 126:761–776. https://doi.org/10.1002/qj.49712656318

Gustafson L (2018) Bayesian tuning and bandits: an extensible, open source library for autoML. M.S. thesis, Dept. of Electrical Engineering and Computer Sciences, Massachusetts Institute of Technology, 100 pp, https://dai.lids.mit.edu/wp-content/uploads/2018/05/Laura_MEng_Final.pdf

Hertel L, Collado J, Sadowski P, Ott J, Baldi P (2020) Sherpa: robust hyperparameter optimization for machine learning. SoftwareX 12:100591. https://doi.org/10.1016/j.softx.2020.100591

Izmailov P, Podoprikhin D, Garipov T, Vetrov D, Wilson AG (2018) Averaging weights leads to wider optima and better generalization. *Conf. on Uncertainty in Artificial Intelligence (UAI) 2018*, Monterey, CA, Association for Uncertainty in Artificial Intelligence, arXiv:1803.05407, https://arxiv.org/abs/1803.05407

Kim PS, Song H-J (2022) Usefulness of automatic hyperparameter optimization in developing radiation emulator in a numerical weather prediction model. Atmosphere 13:721. https://doi.org/10.3390/atmos13050721

Krasnopolsky VM, Fox-Rabinovitz MS, Chalikov DV (2005) New approach to calculation of atmospheric model physics: accurate and fast neural network emulation of longwave radiation in a climate model. Mon Wea Rev 133:1370–1383. https://doi.org/10.1175/MWR2923.1

Krasnopolsky VM, Fox-Rabinovitz MS, Tolman HL, Belochitski AA (2008) Neural network approach for robust and fast calculation of physical processes in numerical environmental models: compound parameterization with a quality control of larger errors. Neural Netw 21:535–543. https://doi.org/10.1016/j.neunet.2007.12.019

Krasnopolsky VM, Fox-Rabinovitz MS, Hou YT, Lord SJ, Belochitski AA (2010) Accurate and fast neural network emulations of model radiation for the NCEP coupled Climate Forecast System: climate simulations and seasonal predictions. Mon Wea Rev 138:1822–1842. https://doi.org/10.1175/2009MWR3149.1

Krasnopolsky VM, Belochitski AA, Hou YT, Lord SJ, Yang F (2012) Accurate and fast neural network emulations of long and short wave radiation for the NCEP Global Forecast System model. NCEP/NWS, NOAA, office note 471, 36 pp, https://repository.library.noaa.gov/view/noaa/6951

Lagerquist R, Turner D, Ebert-Uphoff I, Stewart J, Hagerty V (2021) Using deep learning to emulate and accelerate a radiative-transfer model. J Atmos Oceanic Technol 38:1673–1696. https://doi.org/10.1175/JTECH-D-21-0007.1

Liaw R, Liang E, Nishihara R, Moritz P, Gonzalez JE, Stoica I (2018) Tune: a research platform for distributed model selection and training. *Thirty-fifth International Conference on Machine Learning, AutoML workshop*, Stockholm, Sweden, ICML, arXiv:1807.05118, https://doi.org/10.48850/arXiv.1807.05118

Liu Y, Caballero R, Monteiro JM (2020) RadNet 1.0: exploring deep learning architectures for longwave radiative transfer. Geosci Model Dev 13:4399–4412. https://doi.org/10.5194/gmd-13-4399-2020

Pal A, Mahajan S, Norman MR (2019) Using deep neural networks as cost-effective surrogate models for super-parameterized E3SM radiative transfer. Geophys Res Lett 46:6069–6079. https://doi.org/10.1029/2018GL081646

Roh S, Song H-J (2020) Evaluation of neural network emulations for radiation parameterization in cloud resolving model. Geophys Res Lett 47:e2020GL089444. https://doi.org/10.1029/2020GL089444

Shin H-C, Ha J-H, Ahn KD, Lee EH, Kim CH, Lee YH, Clayton A (2022) An overview of KMA's operational NWP data assimilation system. In: Park SK, Xu L (eds) Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications, Vol IV. Springer, Cham, Switzerland

Skamarock WC, Klemp JB, Dudhia J, Gill DO, Liu Z, Berner J (2019) A description of the Advanced Research WRF model version 4. NCAR Tech. Notes NCAR/TN-556+STR, 148 pp, https://doi.org/10.5065/1DFH-6P97

Song H-J, Kim PS (2022) Effects of cloud microphysics on the universal performance of neural network radiation scheme. Geophys Res Lett 49:e2022GL098601. https://doi.org/10.1029/2022GL098601

Song H-J, Roh S (2021) Improved weather forecasting using neural network emulation for radiation parameterization. J Adv Model Earth Syst 13:e2021MS002609. https://doi.org/10.1029/2021MS002609

Song H-J, Roh S, Park H (2021) Compound parameterization to improve the accuracy of radiation emulator in a numerical weather prediction model. Geophys Res Lett 48:e2021GL095043. https://doi.org/10.1029/2021GL095043

Song H-J, Roh S, Lee J, Nam G, Yun E, Yoon J, Kim PS (2022) Benefits of stochastic weight averaging in developing neural network radiation scheme for numerical weather prediction. J Adv Model Earth Syst 14:e2021MS002921. https://doi.org/10.1029/2021MS002921

Ukkonen P, Pincus R, Hogan RJ, Nielsen KP, Kaas E (2020) Accelerating radiation computations for dynamical models with targeted machine learning and code optimization. J Adv Model Earth Syst 12:e2020MS002226. https://doi.org/10.1029/2020MS002226

Veerman MA, Pincus R, Stoffer R, van Leeuwen CM, Podareanu D, van Heerwaarden CC (2021) Predicting atmospheric optical properties for radiative transfer computations using neural networks. Philos Trans Roy Soc A 379:20200095. https://doi.org/10.1098/rsta.2020.0095

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.