



Geophysical Research Letters

RESEARCH LETTER

10.1029/2020GL089444

Key Points:

- Neural network radiation emulators are developed and evaluated at a cloud-resolving scale
- Radiation emulators with 300 to 56 neurons are 20–100 times faster than the original scheme
- Frequent emulator results can be more accurate than infrequent calculations of the original radiation scheme

Supporting Information:

- Supporting Information S1

Correspondence to:

H.-J. Song,
hwanjinsong@gmail.com

Citation:

Roh, S., & Song, H.-J. (2020). Evaluation of neural network emulations for radiation parameterization in cloud resolving model. *Geophysical Research Letters*, 47, e2020GL089444. <https://doi.org/10.1029/2020GL089444>

Received 21 JUN 2020

Accepted 9 OCT 2020

Accepted article online 17 OCT 2020

Evaluation of Neural Network Emulations for Radiation Parameterization in Cloud Resolving Model

Soonyoung Roh¹ and Hwan-Jin Song¹

¹National Institute of Meteorological Sciences, Korea Meteorological Administration, Jeju, South Korea

Abstract This study evaluated the forecast performance of neural network (NN)-based radiation emulators with 300 to 56 neurons developed under the cloud-resolving simulation. These emulators are 20–100 times faster than the original parameterization and express evolutionary features well for 6 hr. The results suggest that the frequent use of an NN emulator can improve not only computational speed but also forecasting accuracy in comparison to the infrequent use of original radiation parameterization, which is commonly used for speedup but can induce numerical instability as a result of imbalance with other processes. The forecast error of the emulator results was much improved in comparison with that for infrequent radiation runs with similar computational cost. The 56-neuron emulator results were even more accurate than the infrequent runs, which had a computational cost five times higher. The speed and accuracy advantages of radiation emulators can be utilized for weather forecasting.

Plain Language Summary Radiative transfer calculations in weather and climate models often impose computational challenges because of the complexity of radiation processes. Empirical emulators based on NN have been developed to mimic radiation parameterization while reducing computational cost. The accuracy in those studies has not been strictly evaluated because the emulator cannot outpace the original radiation parameterization in terms of accuracy. However, the emulators developed in this study showed advantages both the computational cost and forecast accuracy. These advantages of radiation emulator make them useful for weather forecasting.

1. Introduction

Since radiation is a fundamental energy source that drives weather and climate systems, an appropriate representation of radiative processes plays an important role in weather and climate prediction. The direct approach using a line-by-line radiative model (e.g., Clough et al., 1992, 2005), can compute radiation processes precisely but is prohibitively expensive. To address the computational cost, numerical weather/climate prediction models that employ approximate radiation parameterization (e.g., Baek, 2017; Gu et al., 2011; Iacono et al., 2008) have been developed and heuristically turned to the line-by-line model. Radiation parameterization is still computationally expensive, compared to other schemes, because of the complexity of the underlying physical system. To circumvent the computational cost, radiation parameterizations have been computed less often than the time step of the weather prediction model. However, this approach can lead to significant errors in accumulated discrepancies in interaction with other dynamic/physical processes over time (Pauluis & Emanuel, 2004; Pincus & Stevens, 2013; Xu & Randall, 1995).

The necessity of a trade-off between speed and accuracy in radiation calculations has resulted in the search for alternative approaches, such as a data-driven radiation emulator based on neural networks (NNs), which achieves considerable improvement in speed with reasonable accuracy. Chevallier et al. (1998, 2000) first attempted NN-based longwave (LW) radiation emulation for the European Centre for Medium-Range Weather Forecasts (ECMWF) models. The NN-based LW/shortwave (SW) emulators have been also developed for the Community Atmosphere Model (CAM), the Climate Forecast System (CFS), and the Super-Parameterized Energy Exascale Earth System Model (SP-E3SM) in various studies (Belochitski et al., 2011; Boukabara et al., 2019; Krasnopol'sky et al., 2005; Krasnopol'sky & Fox-Rabinovitz, 2006; Krasnopol'sky, Fox-Rabinovitz, & Belochitski, 2008; Krasnopol'sky, Fox-Rabinovitz, Tolman et al., 2008; Krasnopol'sky et al., 2010; Pal et al., 2019). Krasnopol'sky et al. (2010) presented impressive results for an emulator for the Rapid Radiative Transfer Model for General Circulation Models (RRTMG; Clough et al., 2005;

©2020. The Authors.

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

Iacono et al., 2008), which improved computational speed by 16–60 times in comparison to the original scheme, while preserving long-term (17-yr) stability. In Belochitski et al. (2011), the NN-based radiation emulator provided better performance than the emulators based on the Classification and Regression Tree (CART). Recently, Pal et al. (2019) achieved a tenfold speed improvement and 90–95% accuracy using a deep neural network (DNN), indicating a greater computational burden in DNN. Similarly, various emulators have been developed for idealized frameworks (e.g., Brenowitz & Bretherton, 2018; Krasnopolksy et al., 2013; Rasp et al., 2018) as well as for convection (Gentine et al., 2018), the planetary boundary layer (Wang et al., 2019), and dynamics (Scher, 2018). While the various types of advanced deep-learning techniques that have been recently applied to the postprocessing of weather/climate models, the emulation studies of dynamical/physical processes within the numerical models have remained relatively common techniques regarding the difficulty of application in the numerical model.

Previous radiation emulators were applied to climate simulations at coarse temporal (1–3 hr) and horizontal (100–300 km) resolutions. Although Pal et al. (2019) tried to develop the radiation emulator under the super-parameterized cloud simulation, they applied the results to 1° horizontal resolution. The evaluation of radiation emulator under the cloud-resolving scale (i.e., less than a few km) is essential to be applied to weather forecasting models. However, because the infrequent use of radiation scheme with a substantial speedup is often for rapid forecasting in mesoscale weather prediction models, the radiation emulator is meaningful in weather forecasting when it gives benefits both in speedup and accuracy in comparison to the conventional infrequent radiation runs.

Therefore, this study sought to evaluate the accuracy improvement achieved with a frequently used radiation emulator in comparison to the infrequent original scheme with similar computational cost under the idealized cloud-resolving framework. This evaluation approach is strongly recommended for the future development of radiation emulators. To achieve this goal, we developed an NN-based emulator for radiation parameterization for use with the Korea Local Analysis and Prediction System (KLAPS; Kim et al., 2002), which is an operational short-range weather forecast model used by the Korea Meteorological Administration (KMA). Although this study only involved evaluations in an ideal environment, it is expected that the application of the proposed method to actual weather forecasting will yield many advantages in terms of speedup and accuracy.

2. Training Data and Methods

A two-dimensional idealized squall line simulation was performed with KLAPS, which is based on the Advanced Research Weather Research and Forecasting (WRF-ARW) model (Skamarock et al., 2019). This is a popular cloud simulation for development of microphysics schemes, as well as for understanding cloud-precipitation processes (e.g., Bea et al., 2019; Lim & Hong, 2010; Song et al., 2017). In this experiment, we considered the RRTMG-K radiation (Baek, 2017) and WRF Double Moment 7-Class (WDM7) microphysics (Bea et al., 2019) schemes, which are available in WRF Version 4.1. The RRTMG-K scheme, a two-stream correlated-k approach, optimizes a Monte Carlo independent column approximation and calculates radiative fluxes and heating rates over the LW with 140 g points for 16 bands (within 820–50,000 cm⁻¹) and SW with 112 g points for 14 bands (within 10–3,000 cm⁻¹). The control run was integrated at every model/radiation time step, every 3 s, on 201 horizontal grids (at 0.25-km intervals) and 39 vertical layers (up to 50 hPa) for 6-hr periods (from noon to sunset), which is equivalent to half of a daytime solar cycle. Note that 6 hr is an important regime in the KMA operational short-range forecast system based on the KLAPS and extrapolation of radar observations. The RRTMG-K scheme is responsible for 86.47% of the total computational cost (5,572 s for 6-hr simulation) in the current simulation.

The training sets used to develop the NN emulator were random samples of 20% of the whole data set from the control run (271,360 cases). Although part of the control run included the training sets, this study focused on a limited framework for cloud simulation rather than developing a general NN emulator. For the random sampling, we tried to extract somewhat homogenous distributions for cloud fraction and forecast time, since samplings biased on clear or cloudy cloud conditions or at a specific time cause low accuracy. The NN-emulator inputs for RRTMG-K (196) are as follows: vertical pressure, temperature, water vapor, ozone, cloud fraction, surface temperature, solar constant (G), cosine solar zenith angle ($\cos\theta$), and forecast time (i.e., accumulated time steps). Here, the forecast time was considered as an input variable to alleviate the

error accumulation over the prognostic time. The microphysics variables (cloud liquid/ice/snow effective radius and water path) were excluded from the input data to improve the computational speed and prognostic stability (or implicitly considered), although coupling between radiation and microphysics schemes was inherently allowed (Bea et al., 2016; Bea & Park, 2019). The outputs (86) consist of heating rate profiles and fluxes for LW and SW (Table S1 in the supporting information). For the LW fluxes, there were total-/clear-sky upward fluxes at the top of the atmosphere (TOA) and the surface, respectively, and total-/clear-sky downward fluxes at the surface. Total-sky upward SW fluxes at the TOA and surface were also considered. However, the total-/clear-sky downward SW fluxes at the TOA were directly expressed by $G \times \cos\theta$, and clear-sky upward SW fluxes at the TOA and surface were expressed by linear regressions with respect to $G \times \cos\theta$ because of their strong dependency on the solar cycle. The total-/clear-sky downward SW fluxes at the surface were replaced by total-/clear-sky upward SW fluxes divided by a constant surface albedo (0.2). In addition to surface albedo and emissivity, these replaced flux components should be included in real-case training in the future. Additional redundant constant variables (e.g., trace gases and aerosols) were excluded to avoid additional noise.

The single-layer NN software described by Krasnopolksy (2014) was used in this study to develop the RRTMG-K emulator. For any given inputs, the NN emulator provides approximated outputs without the use of the complex processes in the original parameterization. The approximating function (Equation 1) and related coefficients are learned from the training sets.

$$Y_q = B2_q + \sum_{j=1}^k W2_{qj} \cdot \tanh(B1_j + \sum_{i=1}^n W1_{ji} \cdot X_i), \quad q = 1, 2, \dots, m. \quad (1)$$

Here, n and m indicate the number of inputs and outputs; X_i and Y_q denote the input and output vectors; $W1$ and $W2$ are the matrices of the weights from input to hidden layers [n, k] and from hidden to output layers [k, m], respectively; and $B1$ and $B2$ indicate the bias vectors from input to hidden layers and hidden to output layers, respectively. Here, vertical dependencies between input and output variables are adjusted with the weight and bias coefficients. The accuracy of NN emulation can be tuned by increasing the number of hidden neurons (k), whereas its speedup is inversely proportional to the numerical complexity; $k \times (n + m + 1) + m$, as given by Krasnopolksy et al. (2010). Since the computational cost is increased by k in the numerical complexity, the deep hidden layers such as Pal et al. (2019) may not always be a promising way in terms of speedup though it provides more opportunities for optimization. The weight and bias coefficients are implemented in the NN emulator. The NN emulator replaces combined LW and SW radiations all at once, not separately, and hence, it has an advantage in speedup related to the reduction of the $W1$ and $B1$ arrays, because LW and SW radiations share the majority of inputs. The NN emulators with 300 and 56 neurons (hereinafter referred to as NN300 and NN56) were applied to the 6-hr simulation, which corresponds to 7,200 accumulated model/radiation time steps at increments of 3 s. In terms of computational cost, frequent uses (i.e., every time step) of NN300 and NN56 emulators are equivalent to infrequent uses of the original radiation scheme by 20 and 100 times (WRF20 and WRF100, with 60- and 300-s radiation time steps, respectively). In other words, the calculation of radiation processes in NN300 and WRF20 (NN56 and WRF100) are 20 (100) times faster, compared to the control run with 3-s interval. That is the reason why we considered 56 and 300 neurons in this study. We did not perform the results with smaller neurons than 56 and larger neurons than 300 because the former case could be inaccurate and the latter case has no significance in terms of speedup. The accuracy of these simulations was evaluated by considering the WRF control run to be true. More details related to the development of radiation emulators are described in the supporting information.

3. Results

The trained heating rate and flux results are shown in Figure 1. Although the NN training was designed to identify an optimized convergence solution for all given input-output pairs, the explanation obtained from the inputs may vary depending on the characteristics of the outputs. We note that the training results (in terms of R^2) for the heating rate profiles are less accurate than those for the single-level fluxes, implying that vertical profiles involve greater uncertainty than single-level products. The SW results exhibit lower R^2 and higher the root mean square error (RMSE) for fluxes than those of the LW. These results suggest that SW processes are more complex than LW processes at the cloud-resolving scale. Some of the uncertainty is

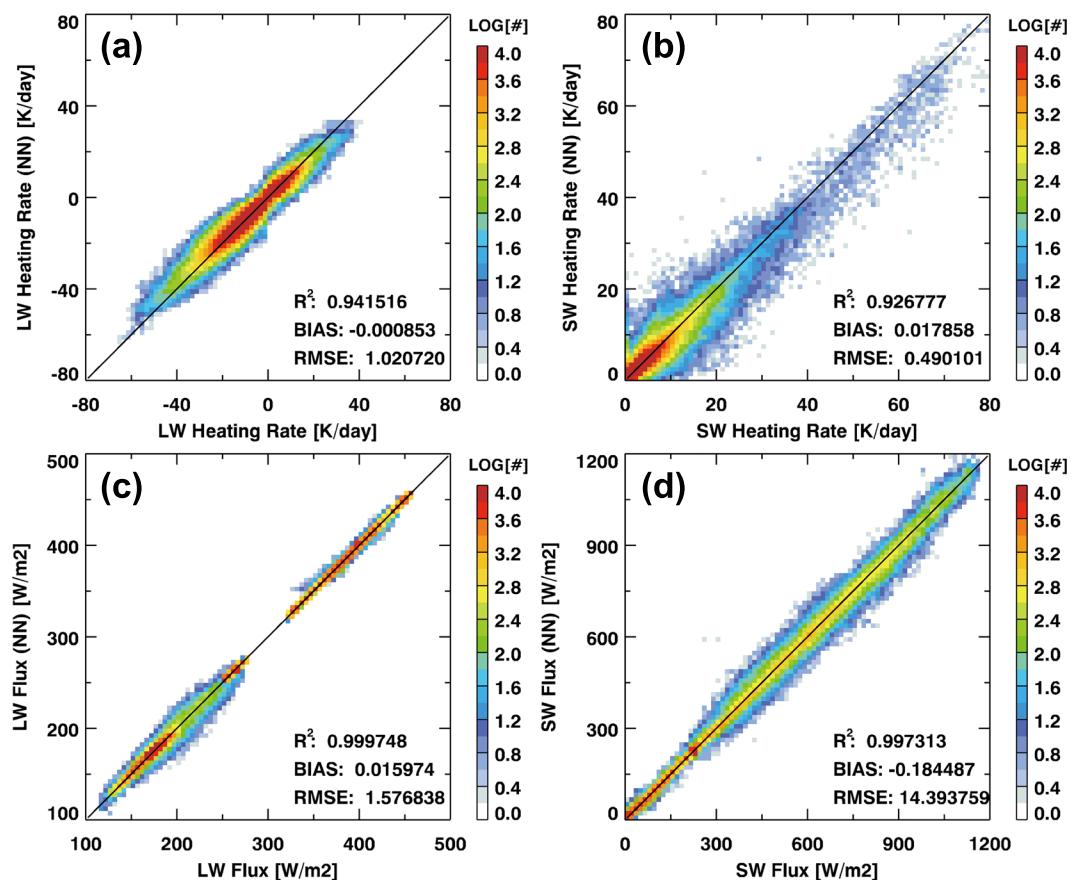


Figure 1. Comparison of (a) LW heating rate, (b) SW heating rate, (c) LW flux, and (d) SW flux between the control run and NN emulation with 300 neurons for training data sets. Heating rates for 39 vertical layers and six LW and two SW fluxes are expressed together in the figure. The colors in the figure represent the occurrence frequency on a log scale.

presumed to be related to the excluded microphysics variables (when the microphysics variables were further considered in training sets, RMSEs for SW and LW processes were reduced by 26% and 12%, respectively). It also notes that RMSE of the 10% independent test data (on the average of four RMSEs in Figure 1) was increased by 26% compared with the training sets (Table S2).

Figure 2 shows the vertical distribution of the horizontal (50 km) mean cloud fraction and heating rate (LW and SW) with the accumulated forecast time for the WRF control run, NN300, and WRF20. The experiment simulated a vertically developing cloud by initial forcing of warm bubble heating at the lower center of the domain, following precipitation. Hence, the cloud fraction as a key factor in determining radiative processes in this experiment. Until approximately 20 min had elapsed, the negative LW and positive SW values were clearly detected as the cloud grew to 9 km (Figures 2b and 2c). Although initial cloud forcing occurred mainly near the center of the domain, it was also identified in the 50-km mean feature because of its strength. The cloud top, regarded as a 10% cloud fraction, developed up to 12 km for 2 hr, but after that, the cloud top height decreased to 9 km (Figure 2a). Areas with more than 90% cloud fractions were present at an altitude of approximately 9 km on average at 2 hr but lowered further to approximately 6.5 km after 6 hr. Evolutionary features of the LW and SW heating rate profiles, the main outputs of the radiation parameterization, are shown in Figures 2b and 2c. Above the high cloud fraction, the strongest LW cooling area was located over 9–12 km for 3 hr but fell to 7–9 km within 6 hr (Figure 2b). The strongest SW warming was also found over the LW cooling area for the first 3 hr (Figure 2c). Similar LW cooling and SW warming feature responses to the cloud fraction were reported by Zhang et al. (2017). Although the LW cooling trend lasted up to 6 hr, the SW warming weakened rapidly after 3 hr because of reduced solar insolation at an increased zenith angle. Weak LW warming also appeared below the cloud layer and near the surface after 2 hr.

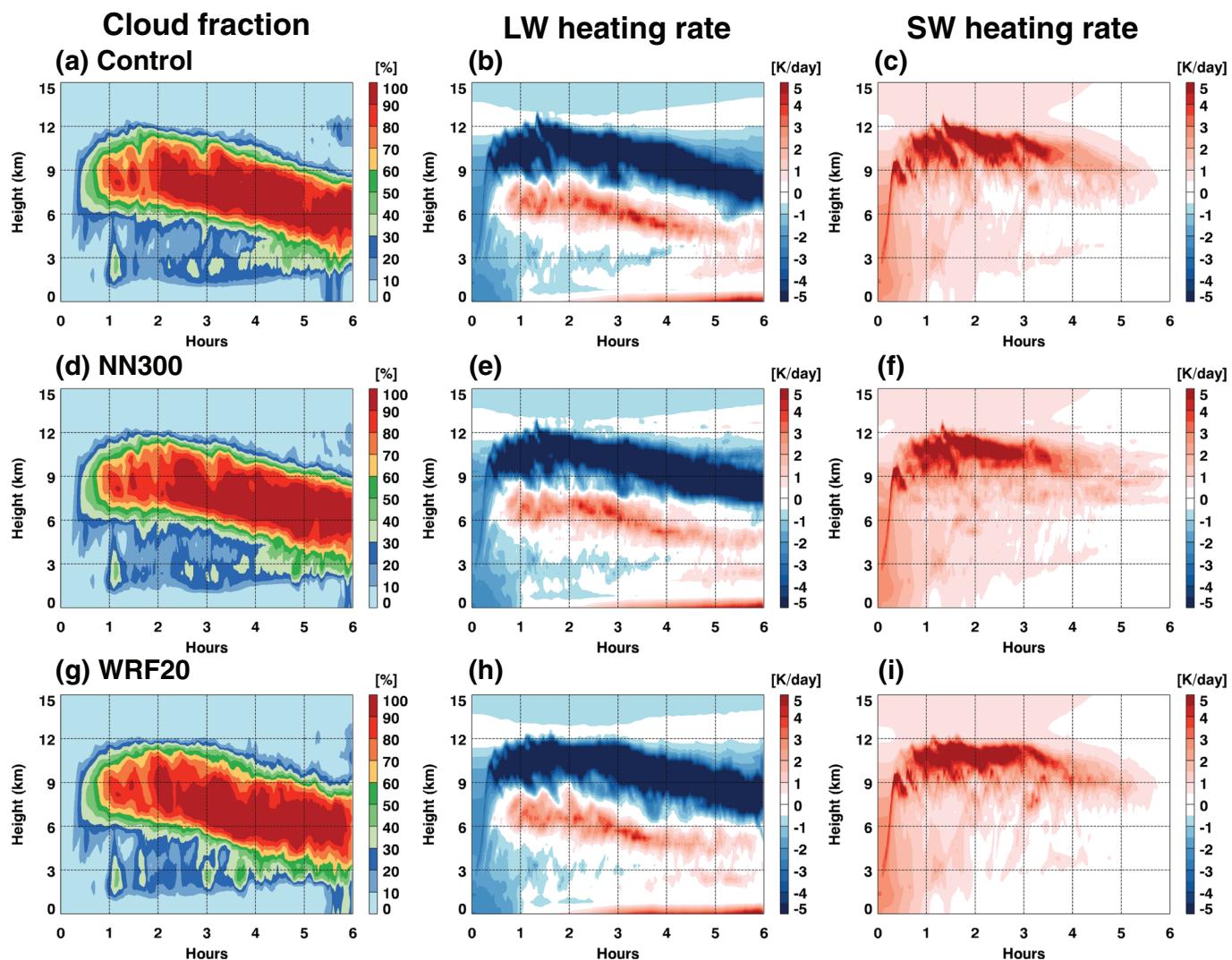


Figure 2. Temporal variation of 50-km mean (a) cloud fraction and (b) LW and (c) SW heating rate profiles for the control run. (d–f) The same as (a)–(c) but for the WRF simulation using the NN emulator with 300 neurons (NN300). (g–i) The same as (a)–(c) but for the WRF simulation with the infrequent radiation time step by 20 times (WRF20).

The evolutionary feature of NN300, equivalent to 20 times speedup, exhibited good agreement with the control run, even during the latter part of the forecast time (middle row of Figure 2). Concerning the upward development of clouds, the LW cooling and SW warming features within the first 30 min were accurately simulated (Figures 2e and 2f). The strong LW cooling area present over 9–12 km within 3 hr and 7–9 km at 6 hr was well represented in NN300 (Figure 2e). The weak LW warming below the cloud and over the surface (Figure 2e) and the strong SW warming area above 3 K day^{-1} (Figure 2f) were also represented as well. The WRF20 exhibited similar performance to the NN300 for large-scale features (Figures 2g–2i). However, Table 1 shows that the NN300 results agreed better with the control run (i.e., lower RMSE and higher R^2) than did the WRF20 results. Note that the mean biases were both close to zero, so the RMSE results may be the most appropriate measure of accuracy. In terms of the RMSE, the accuracy of NN300 was improved by 19% for the LW heating rate, 22% for the SW heating rate, and 25% for the cloud fraction, relative to WRF20. The NN56, equivalent to a hundredfold speedup, yielded RMSE improvements of 24% for the LW heating rate, 11% for the SW heating rate, and 42% for the cloud fraction, in comparison to WRF100. Surprisingly, the NN56 results were even more accurate than the WRF20, despite the huge difference in computational cost during the 6-hr forecast period (i.e., a factor of 100 vs. 20 in speedup). These results suggest that the frequent use of a radiation emulator can be beneficial in terms of both computational speed and

Table 1

Evaluations Results for NN Emulators Using 300 and 56 Neurons (NN300 and NN56) and Infrequent Radiation Time Steps by 20 and 100 Times (WRF20 and WRF100) Versus the WRF Control Run

Experiments	NN300	WRF20	NN56	WRF100
Speedup of radiation	20.76	20	100.81	100
Reduced computation time	82.12%	82.03%	85.96%	85.64%
LW heating rate (K day^{-1})	0.92 (0.92)	1.14 (0.88)	1.03 (0.90)	1.35 (0.83)
SW heating rate (K day^{-1})	0.40 (0.90)	0.51 (0.84)	0.47 (0.86)	0.53 (0.83)
Cloud fraction (%)	6.04 (0.97)	8.04 (0.95)	6.27 (0.97)	10.86 (0.91)
LW flux (W m^{-2})	5.68 (1.00)	7.97 (0.99)	6.15 (1.00)	8.03 (0.99)
LWUPT	10.90 (0.91)	13.55 (0.87)	11.49 (0.87)	15.13 (0.84)
LWUPTC	1.86 (0.86)	2.85 (0.71)	2.18 (0.80)	3.22 (0.71)
LWUPB	1.03 (0.85)	1.56 (0.68)	1.13 (0.81)	1.41 (0.76)
LWUPBC	0.72 (0.97)	1.31 (0.89)	0.96 (0.91)	1.12 (0.94)
LWDNB	6.84 (0.84)	10.42 (0.66)	7.19 (0.82)	9.42 (0.74)
LWDNBC	4.78 (0.97)	8.76 (0.88)	6.06 (0.92)	7.44 (0.94)
SW flux (W m^{-2})	38.58 (0.99)	48.52 (0.99)	46.79 (0.98)	59.36 (0.97)
SWUPT	53.27 (0.97)	66.70 (0.95)	64.76 (0.95)	81.59 (0.93)
SWUPB	11.78 (0.96)	16.11 (0.93)	13.62 (0.94)	19.74 (0.90)
Surface temperature (T_s) (K)	0.92 (0.94)	1.62 (0.83)	1.07 (0.92)	1.78 (0.88)
Precipitation (mm)	0.19 (0.58)	0.24 (0.36)	0.18 (0.58)	0.21 (0.51)

Note. The statistics in the table are based on results in Figures 2 and 3 and represent the root mean square error (RMSE) and the square of the correlation coefficient (in parentheses) relative to the control run.

accuracy, relative to the infrequent use of the original scheme, especially for severe weather forecasting for which radiative processes at the cloud-resolving scale are important.

The upper panel of Figure 3 shows evolutionary features in the horizontal domain (x) for LW/SW fluxes, surface temperature (T_s), and precipitation for the control run. The total-sky LW upward flux at the TOA (LWUPT) exhibited a high value under the clear sky in the early stages but rapidly decreased in relation to horizontally spread clouds (Figure 2a). Unlike the LWUPT, total-sky SW upward fluxes at the TOA (SWUPT) were greatly increased by cloudy conditions but then gradually decayed until sunset. The LW and SW fluxes at the surface (LWUPB, LWUPBC, LWDNB, LWDNBC, and SWUPB) developed into a horizontally asymmetric pattern tilted in the positive x direction that decreased toward sunset. These features are thought to be intimately related to T_s . Precipitation was mainly distributed over the ± 10 -km area corresponding to the center of the clouds and appears to have been biased toward the negative x direction after 5 hr.

Both NN300 and WRF20 represent characteristic features found in the control run, although difference exists from point to point (Figure 3). Although the developed NN emulator inevitably includes the discrepancies, their differences are within reasonable limits, as listed in Table 1. Here, statistics in Table 1 are obtained every 1 min. The reduced computational cost by 82–86% results from the comparison with the WRF control run (i.e., 5,572 s). The NN300 exhibited improvements of 28% in LW fluxes and 20% in SW fluxes, in terms of RMSE, compared to the WRF20. The improvements were mainly associated with LWUPT, LWDNB, LWDNBC, SWUPT, and SWUPB, with the results for these exhibiting the largest discrepancies for the control run. In particular, NN56 showed a 23% lower RMSE, compared to the LW fluxes of WRF20, implying advantages in both speed and accuracy. The NN300 reasonably simulated areas with T_s greater than 298 K for up to 3 hr as well, but WRF20 does not provide this feature. Compared to WRF20, the T_s results from NN300 and NN56 represent significant reductions in RMSE, that is, 43% and 34%, respectively, as well as more accurate pattern correlations, that is, 0.94 and 0.92, respectively. Precipitation is a bottleneck in prognostic forecasting since it is difficult to simulate accurately under conditions of higher uncertainty and complexity (relatively lower R^2 values in Table 1). Nevertheless, the NN300 results effectively represented the precipitation pattern, which was mainly concentrated at ± 10 km in the control run (Figure 3). However, WRF20 exhibited a heavy rainfall area up to the 20-km point, resulting in a huge difference with respect to the control run, especially after 4 hr. For precipitation, the NN300 and NN56 results represented improvements of 25% and 21% in terms of reduced RMSE and enhanced R^2 (0.58), in comparison to the WRF20.

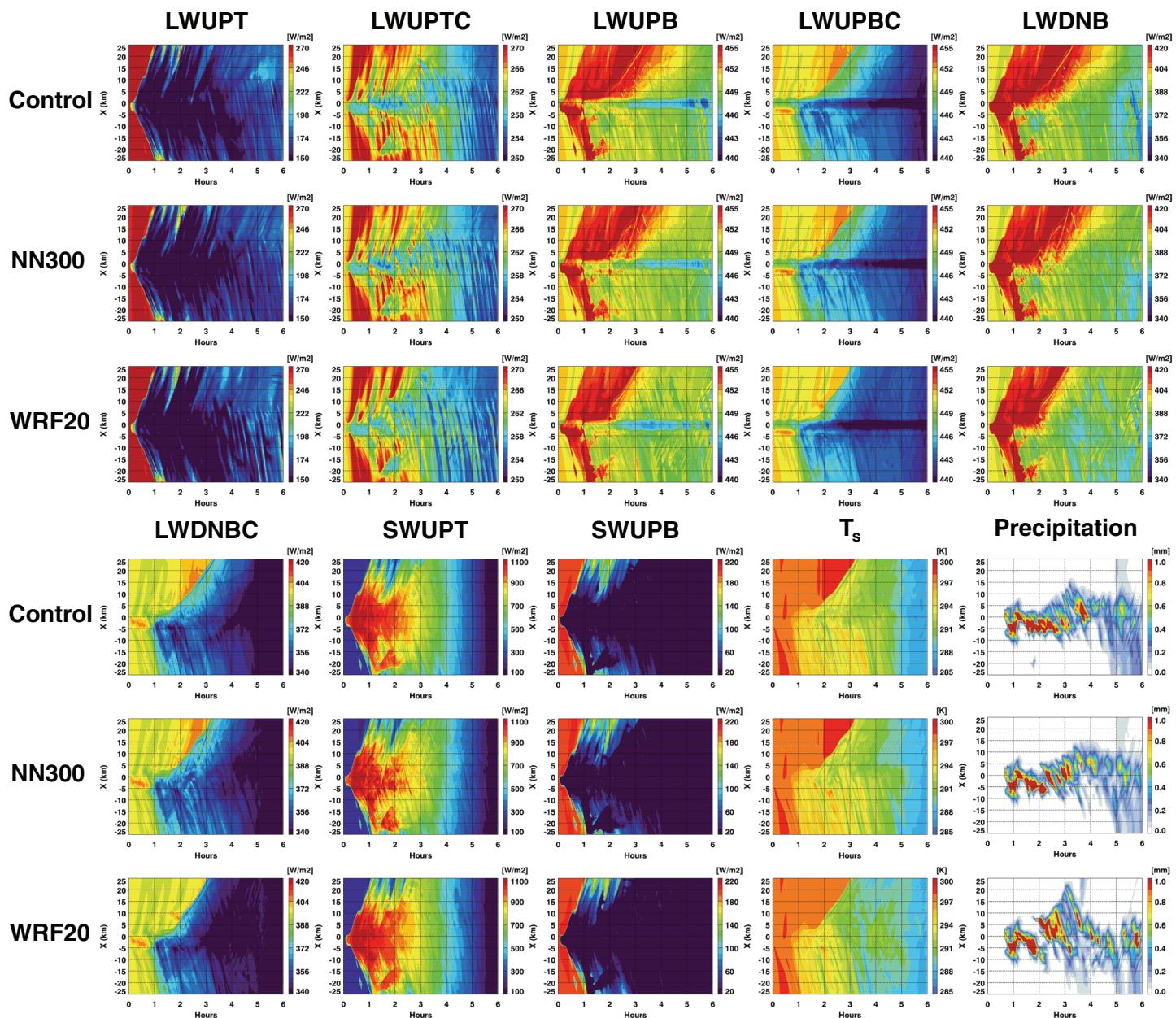


Figure 3. Temporal and spatial variation of LW and SW fluxes, surface temperature (T_s), and precipitation for the control run (top panel), NN300 (middle panel), and WRF20 (bottom panel). LW, SW, UP, DN, T, B, and C indicate longwave, shortwave, upward, downward, top of atmosphere, bottom of atmosphere, and clear sky, respectively.

Note that this study did not include particular parameter tuning besides the default recommendation by Krasnopolsky (2014), further enhancing of accuracy could be possible by parameter optimization (supporting information). Several sensitivity tests with different hidden neurons, activation functions, and initial soundings are shown in the supporting information. Although the accuracy for training (or test) sets tends to be generally improved as neurons increases (Table S2), the performance with those neurons appears to be saturated at a certain level (Table S3), implying the emulator with smaller neurons could be beneficial in terms of both speed and accuracy. All NN-emulator results in Table S3 are more accurate than the WRF20, indicating further advantage in computational cost. Further consideration of deep hidden layers and different neurons (more or less than this study) will provide more opportunities to find the best solution for improving both accuracy and speedup. Table S4 exhibits that the use of activation functions (sigmoid,

softsign, and arctan) with nonlinear property produce better performance compared with the linear function, while such nonlinear functions still present inaccurate results rather than tanh. Note that most radiation emulators in literatures and this study have been developed with tanh, except for sigmoid in Pal et al. (2019). However, optimal activation function may vary depending on the situation. The sensitivity experiments with different initial soundings were further performed using cold- and warm-type heavy rain perturbations reported in Song and Sohn (2015) and Song et al. (2017). Here, the cold- and warm-type experiments represent thermodynamically more unstable and stable environments, respectively. The RMSE and pattern correlation of the cold type were lower than those of warm type. It is thought to be related to the growth of the nonlinearity and instability by more cloudy conditions in the cold type. However, both the cold- and warm-type experiments using 56-neurons emulator represented a considerable improvement in accuracy compared to the same computational cost simulation (WRF100).

4. Summary and Conclusion

In order to evaluate the performance of NN-based radiation emulator at the cloud-resolving scale, RRTMG-K radiation emulators were developed with 300 to 56 neurons (NN300 to NN56) and implemented in the WRF model in the framework of an idealized squall-line simulation with 250-m spacing and 3-s time step. A combined algorithm for LW and SW radiation was introduced for speedup and was integrated over 6 hr. The emulator results with 20–100 times speedup appeared to reproduce well vertical evolutionary features of the LW/SW heating rate related to the cloud fraction. The prognostic features of LW/SW fluxes, surface temperature, and precipitation were also well simulated with the emulators. The NN300 and NN56 results, applied every time step, were compared with infrequent uses of 60- and 300-s radiation time steps (WRF20 and WRF100), equivalent to the same computational cost. In evaluation with the control simulation every radiation time step, the accuracy improvement achieved with NN300 (NN56), in terms of RMSE, were 19% (24%) for LW heating rate, 22% (11%) for SW heating rate, 25% (42%) for cloud fraction, 28% (23%) for LW fluxes, 20% (16%) for SW fluxes, 43% (34%) for surface temperature, and 25% (21%) for precipitation, compared to those obtained with WRF20 and WRF100, respectively. The NN56 results were even more accurate than the WRF20 results, despite 80% lower computational cost. Several experiments with different hidden neurons, activation functions, and initial soundings were further performed.

Since previous radiation emulators have applied to the climate simulations at horizontal resolutions of 100–300 km, these results are particularly meaningful in terms of first attempt to evaluate the performance of radiation emulators at the cloud-resolving scale with strong nonlinearity. It is essential to forecasting severe weather accompanied by complex cloud systems. Furthermore, the evaluation method developed in this study (which achieves better performance than the infrequent use of the original radiation scheme, equivalent to similar speedup condition) provides insights that will be useful in the future development of radiation emulators. This is important because an emulator should provide benefits in speedup or accuracy in comparison to the infrequent radiation method. No previous research has presented an evaluation of radiation emulator under such strict condition, because previous radiation emulators were focused on imitating the original radiation scheme in climate simulations with coarse temporal resolutions (1–3 hr). We acknowledge that the results obtained in this study apply to very limited ideal condition that cannot be easily generalized when applied to an actual case. Therefore, it will be necessary to ensure that the forecast performance (especially for severe weather) is truly improved by applying NN-based radiation emulators to real cases in the future.

Data Availability Statement

The WRF model used in this study was obtained online (from <https://www2.mmm.ucar.edu/wrf/users>).

References

- Baek, S. (2017). A revised radiation package of G-packed McICA and two-stream approximation: Performance evaluation in a global weather forecasting model. *Journal of Advances in Modeling Earth Systems*, 9, 1628–1640. <https://doi.org/10.1002/2017MS000994>
- Bea, S. Y., Hong, S.-Y., & Lim, K. S. S. (2016). Coupling WRF Double-Moment 6-Class microphysics schemes to RRTMG radiation scheme in Weather Research Forecasting model. *Advances in Meteorology*, 2016, 1–11. <https://doi.org/10.1155/2016/5070154>
- Bea, S. Y., Hong, S.-Y., & Tao, W.-K. (2019). Development of a single-moment cloud microphysics scheme with prognostic hail for the Weather Research and Forecasting (WRF) model. *Asia-Pacific Journal of Atmospheric Sciences*, 55, 233–245. <https://doi.org/10.1007/s13143-018-0066-3>

Acknowledgments

We are deeply grateful to Vladimir Krasnopolksy and Alexei Belochitski (NOAA) for providing the Fortran-based NN software used in this study and their constructive comments. We also acknowledge the dedicated efforts of Vijay Tallapragada, Hyesook Lee, and Yong Hee Lee for planning the international joint research between NOAA and KMA. This work was funded by the KMA Research and Development Program “Development of the AI technique for the prediction of rainfall optimized over the Korean peninsula” under Grant (KMA2018-00124).

- Bea, S. Y., & Park, R.-S. (2019). Consistency between the cloud and radiation processes in a numerical forecasting model. *Meteorology and Atmospheric Physics*, 131, 1429–1436. <https://doi.org/10.1007/s00703-018-0647-9>
- Belochitski, A., Binev, P., DeVore, R., Fox-Rabinovitz, M., Krasnopol'sky, V., & Lamby, P. (2011). Tree approximation of the long wave radiation parameterization in the NCAR CAM global climate model. *Journal of Computational and Applied Mathematics*, 236, 447–460. <https://doi.org/10.1016/j.cam.2011.07.013>
- Boukabara, S. A., Krasnopol'sky, V., Stewart, J. Q., Maddy, E. S., Shahroodi, N., & Hoffman, R. N. (2019). Leveraging modern artificial intelligence for remote sensing and NWP: Benefits and challenges. *Bulletin of the American Meteorological Society*, 100, 473–491. <https://doi.org/10.1175/BAMS-D-18-0324.1>
- Brenowitz, N. D., & Bretherton, C. S. (2018). Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, 45, 6289–6298. <https://doi.org/10.1029/2018GL078510>
- Chevallier, F., Chéruy, F., Scott, N. A., & Chédin, A. (1998). A neural network approach for a fast and accurate computation of a longwave radiative budget. *Journal of Applied Meteorology*, 37(11), 1385–1397. [https://doi.org/10.1175/1520-0450\(1998\)037<1385::AONNAF>2.0.CO;2](https://doi.org/10.1175/1520-0450(1998)037<1385::AONNAF>2.0.CO;2)
- Chevallier, F., Morcrette, J.-J., Chéruy, F., & Scott, N. A. (2000). Use of a neural-network-based long-wave radiative-transfer scheme in the ECMWF atmospheric model. *Quarterly Journal of the Royal Meteorological Society*, 126(563), 761–776. <https://doi.org/10.1002/qj.49712656318>
- Clough, S. A., Iacono, M. J., & Moncet, J.-L. (1992). Line-by-line calculation of atmospheric fluxes and cooling rates: Application to water vapor. *Journal of Geophysical Research*, 97(D14), 15,761–15,785. <https://doi.org/10.1029/92JD01419>
- Clough, S. A., Shephard, M. W., Mlawer, E. J., Delamere, J. S., Iacono, M. J., Cady-Pereira, K., et al. (2005). Atmospheric radiative transfer modeling: A summary of the AER codes. *Journal of Quantitative Spectroscopy & Radiative Transfer*, 91, 233–244. <https://doi.org/10.1016/j.jqsrt.2004.05.058>
- Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., & Yacalis, G. (2018). Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, 45, 5742–5751. <https://doi.org/10.1029/2018GL078202>
- Gu, Y., Liou, K. N., Ou, S. C., & Fovell, R. (2011). Cirrus cloud simulations using WRF with improved radiation parameterization and increased vertical resolution. *Journal of Geophysical Research*, 116, D06119. <https://doi.org/10.1029/2010JD014574>
- Iacono, M. J., Delamere, J. S., Mlawer, E. J., Shephard, M. W., Clough, S. A., & Collins, W. D. (2008). Radiative forcing by long-lived greenhouse gases: Calculations with the AER radiative transfer models. *Journal of Geophysical Research*, 113, D13103. <https://doi.org/10.1029/2008JD009944>
- Kim, Y.-S., Park, O.-R., & Hwang, S.-O. (2002). Realtime operation of the Korea Local Analysis and Prediction System at METRI. *Asia-Pacific Journal of Atmospheric Sciences*, 38, 1–10.
- Krasnopol'sky, V. (2014). NCEP neural network training and validation system: Brief description of NN background and training software. Environment Modeling Center, NCEP/NWS, NOAA, <https://doi.org/10.7289/v5qr4v2z>
- Krasnopol'sky, V., & Fox-Rabinovitz, M. S. (2006). Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. *Neural Networks*, 19(2), 122–134. <https://doi.org/10.1016/j.neunet.2006.01.002>
- Krasnopol'sky, V., Fox-Rabinovitz, M. S., & Belochitski, A. A. (2008). Decadal climate simulations using accurate and fast neural network emulation of full, longwave and shortwave, radiation. *Monthly Weather Review*, 136, 3683–3695. <https://doi.org/10.1175/2008MWR2385.1>
- Krasnopol'sky, V., Fox-Rabinovitz, M. S., & Belochitski, A. A. (2013). Using ensemble of neural networks to learn stochastic convection parameterizations for climate and numerical weather prediction models from data simulated by a cloud resolving model. *Advances in Artificial Neural Systems*, 2013, 485913. <https://doi.org/10.1155/2013/485913>
- Krasnopol'sky, V., Fox-Rabinovitz, M. S., & Chalikov, D. V. (2005). New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of longwave radiation in a climate model. *Monthly Weather Review*, 133, 1370–1383. <https://doi.org/10.1175/MWR2923.1>
- Krasnopol'sky, V., Fox-Rabinovitz, M. S., Hou, Y. T., Lord, S. J., & Belochitski, A. A. (2010). Accurate and fast neural network emulations of model radiation for the NCEP coupled Climate Forecast System: Climate simulations and seasonal predictions. *Monthly Weather Review*, 138(5), 1822–1842. <https://doi.org/10.1175/2009MWR3149.1>
- Krasnopol'sky, V., Fox-Rabinovitz, M. S., Tolman, H. L., & Belochitski, A. A. (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 Networks*, 21(2–3), 535–543. <https://doi.org/10.1016/j.neunet.2007.12.019>
- Lim, K.-S. S., & Hong, S.-Y. (2010). Development of an effective double-moment cloud microphysics scheme with prognostic Cloud Condensation Nuclei (CCN) for weather and climate models. *Monthly Weather Review*, 138(5), 1587–1612. <https://doi.org/10.1175/2009MWR2968.1>
- Pal, A., Mahajan, S., & Norman, M. R. (2019). Using deep neural networks as cost-effective surrogate models for super-parameterized E3SM radiative transfer. *Geophysical Research Letters*, 46, 6069–6079. <https://doi.org/10.1029/2018GL081646>
- Pauluis, O., & Emanuel, K. (2004). Numerical instability resulting from infrequent calculation of radiative heating. *Monthly Weather Review*, 132, 673–686. [https://doi.org/10.1175/1520-0493\(2004\)132<673:NIRFIO>2.0.CO;2](https://doi.org/10.1175/1520-0493(2004)132<673:NIRFIO>2.0.CO;2)
- Pincus, R., & Stevens, B. (2013). Paths to accuracy for radiation parameterizations in atmospheric models. *Journal of Advances in Modeling Earth Systems*, 5, 225–233. <https://doi.org/10.1002/jame.20027>
- Rasp, S., Pritchard, M. S., & Gentile, P. (2018). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115, 9684–9689. <https://doi.org/10.1073/pnas.1810286115>
- Scher, S. (2018). Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning. *Geophysical Research Letters*, 45, 12,616–12,622. <https://doi.org/10.1029/2018GL080704>
- Skamarock, W. C., Klemp, J. B., Dudhia, J., Gill, D. O., Liu, Z., Berner, J., et al. (2019). A description of the advanced research WRF model Version 4, NCAR Technical Notes. <https://doi.org/10.5065/1DFH-6P97>
- Song, H.-J., & Sohn, B.-J. (2015). Two heavy rainfall types over the Korean Peninsula in the humid East Asian summer environment: A satellite observation study. *Monthly Weather Review*, 143(1), 363–382. <https://doi.org/10.1175/mwr-d-14-00184.1>
- Song, H.-J., Sohn, B. J., Hong, S.-Y., & Hashino, T. (2017). Idealized numerical experiments on the microphysical evolution of warm-type heavy rainfall. *Journal of Geophysical Research: Atmospheres*, 122, 1685–1699. <https://doi.org/10.1002/2016JD02563>
- Wang, J., Balaprakash, P., & Kotamarthi, R. (2019). Fast domain-aware neural network emulation of a planetary boundary layer parameterization in a numerical weather forecast model. *Geoscientific Model Development*, 12, 4261–4274. <https://doi.org/10.5194/gmd-12-4261-2019>

- Xu, K.-M., & Randall, D. A. (1995). Impact of interactive radiative transfer on the macroscopic behavior of cumulus ensembles. Part I: Radiation parameterization and sensitivity tests. *Journal of the Atmospheric Sciences*, 52(7), 785–799. [https://doi.org/10.1175/1520-0469\(1995\)052<785>2.0.CO;2](https://doi.org/10.1175/1520-0469(1995)052<785>2.0.CO;2)
- Zhang, K., Randel, W. J., & Fu, R. (2017). Relationships between outgoing longwave radiation and diabatic heating in reanalyses. *Climate Dynamics*, 49, 2911–2929. <https://doi.org/10.1007/s00382-016-3501-0>



Geophysical Research Letters

Supporting Information for

**Evaluation of neural network emulations for radiation parameterization in cloud
resolving model**

Soonyoung Roh and Hwan-Jin Song

National Institute of Meteorological Sciences
Korea Meteorological Administration
Jeju-do, Republic of Korea

Contents of this file

Text S1

Figures S1 to S2

Tables S1 to S5

Introduction

This supporting information provides additional contents (Text S1, Figures S1–2, and Table 1) for describing the whole processes for the development of neural network radiation emulator, as well as sensitivity tests using different hidden neurons, activation functions, and initial soundings (Tables S2 to S5).

Text S1

Step 1: Prepare the control run and input-output sets

In order to develop the radiation emulator by completely replacing a certain radiation parameterization scheme, input and output datasets need to be prepared first. If conventional outputs are used, there is a possibility of data contamination related to the order of dynamic/physical processes. For example, cloud fraction is the primary input of radiation parameterization; however, it is also a conventional product affected by sequential order within a numerical model. In operational products (e.g., generated with hourly periods) are considered inputs and outputs for developing the emulator, they may not be accurate inputs and outputs for a particular physics scheme at a given time step (usually several seconds in weather forecasting model). For this reason, the first step is to modify the WRF codes (written in Fortran) to write the input and output (Figure S1). Here, Registry.EM_COMMON is associated with the variable declaration (including the variable names commonly used in the WRF, as well as new input-output variables for NN training), and module_radiation_driver.F and first_rk_step_part1.F correspond to driver linking the radiation processes to other dynamic and physical processes. The main code of shortwave (SW) parameterization is module_ra_rrtmg_swk.F. As the longwave (LW) parameterization was a subroutine of the RRTMG-K SW code, the modification to specify input-output in the whole processes of RRTMG-K was performed on the SW scheme. After these modifications, idealized two-dimensional squall line experiments (em_squall2d_x) were performed. More details with the WRF model are available at <https://www2.mmm.ucar.edu/wrf/users>. As a result, the input-output sets for NN training were prepared with the NetCDF format for each time step (3 s in this study). Lists of input and output variables are shown in Table S1. The control run for evaluating the performance of the radiation emulator was prepared with default codes without modification in the first step to calculate the exact calculation time as the reference.

Step 2: Preprocessing for training sets

In the second step, the inputs and outputs were rearranged into [# of cases, # of input variables] and [# of cases, # of output variables], respectively. Approximately hundred thousand cases were randomly selected from the whole datasets. In addition, constant input and output variables were removed to reduce errors and save computational cost. This study further used the preprocessing method for NN training as proposed by Krasnolpolsky (2014). This includes normalization for inputs/outputs and weight initialization method (Nguyen and Widrow 1990). Here, inputs were normalized to an interval $[-1, 1]$ using the maximum and minimum values of each input variable using equation (2.6) in Krasnopolksy (2014). Although the use of a different input range may provide greater optimization according to the situation, this study did not attempt the sensitivity experiment. Outputs were converted using traditional normalization based on mean and standard deviation, given in equation (2.8) in Krasnopolksy (2014). The predicted outputs were converted into the original dimension by the inverse-normalization. In this study, the second step was performed in the Interactive Data Language (IDL) codes.

Step 3: NN training

The third step demonstrates the main NN training process performed using Fortran code. In the beginning, the training sets with the learning rates (10^{-3}) and maximum iteration number for optimal convergence was used. Here, the epoch and iteration are the same in this NN training. The NN training includes the early stopping controller. The training process is discontinued if the learning rate is less than 10^{-6} (learning rate is adjusted to one order smaller if the error converge is not enough). Further, the NN training is also stopped if one condition is satisfied within the following criteria: minimal normalized root mean square error (RMSE) $< 10^{-2}$, minimal change in normalized RMSE $< 10^{-5}$, or minimal change in weights $< 10^{-2}$. The detailed control parameters are given in Table 3.2 of Krasnopolksy (2014). Optimal values for the weights were calculated by minimizing the error function. To solve the nonlinear minimizing problem, including a penalty term to avoid over-fitting such as local minima, the present study used a simplified version of the gradient descent method for a back-propagation algorithm. See section 2.3 of Krasnopolksy (2014) for further details.

The NN training was performed with a serial process with the Central Processing Unit (CPU)-based supercomputer systems (Cray XC40, Xeon E5-2690v3 12C 2.6GHz, Aries, interconnect, Cray/HPE) in the Korea Meteorological Administration (called as Miri and Nuri). As shown in Figure S2, the NN training processes with 56 neurons (NN56) and 300 neurons (NN300) were converged after 4,658 and 7,728 iterations, respectively. Whether to converge or not was determined following 5,660 and 8,729 iterations for NN56 and NN300, respectively. The CPU time consumed for the NN training was 37,961 s and 449,866 s for NN56 and NN300, respectively. Weight and bias coefficients were finally obtained as a result of NN training: W1 [196, 300], B1 [300], W2 [300, 86], B2 [86] for 300 neurons. Note that the NN training could be better optimized by using other techniques that parallel training is supported (e.g., python packages: pytorch, scikit-learn, tensorflow+keras).

Step 4: Experiments with radiation emulator

In this step, a total of 25,989 lines of RRTMG-K SW/LW codes were replaced by approximately 200 lines for NN emulator (phys/module_ra_rrtmg_swk.F), implying a significantly faster calculation. All RRTMG-K processes were removed with the exclusion of the reading part of input variables and the writing part of output variables. It was completely replaced by the radiation emulator based on weight and bias coefficients from the NN training. Finally, idealized squall line experiments are performed using the radiation emulator and the results are compared with the control run.

Krasnopolksy, V. (2014). NCEP neural network training and validation system: Brief description of NN background and training software. *Environment Modeling Center, NCEP/NWS, NOAA*, <https://doi.org/10.7289/v5qr4v2z>.

Nguyen, D., and Widrow, B. (1990) Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. *Proc of International Joint Conference of Neural Networks*, June 17–21, San Diego, CA, USA, 3:21–26.

Skamarock, W. C., Klemp, J. B., Dudhia, J., Gill, D. O., Liu, Z., Berner, J., Wang, W., Powers, J. G., Duda, M. G., Barker, D. M., and Huang, X.-Y. (2019). A description of the Advanced Research WRF model version 4, *NCAR Technical Notes*, <https://doi.org/10.5065/1DFH-6P97>.

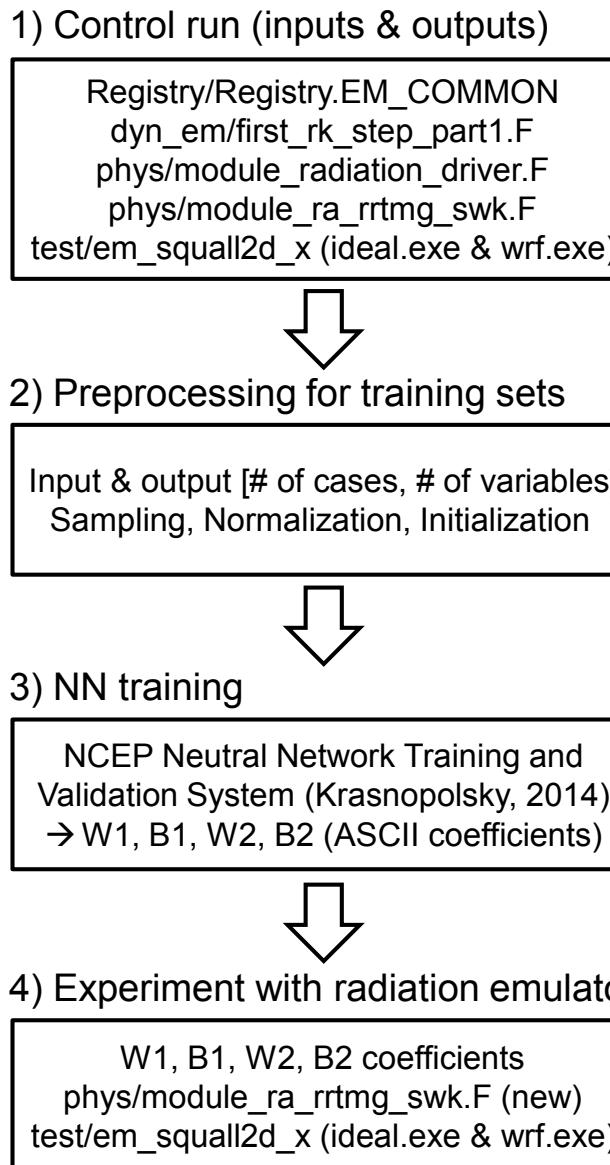


Figure S1. Schematic diagram representing the whole process for developing the neural network (NN)-based radiation emulator.

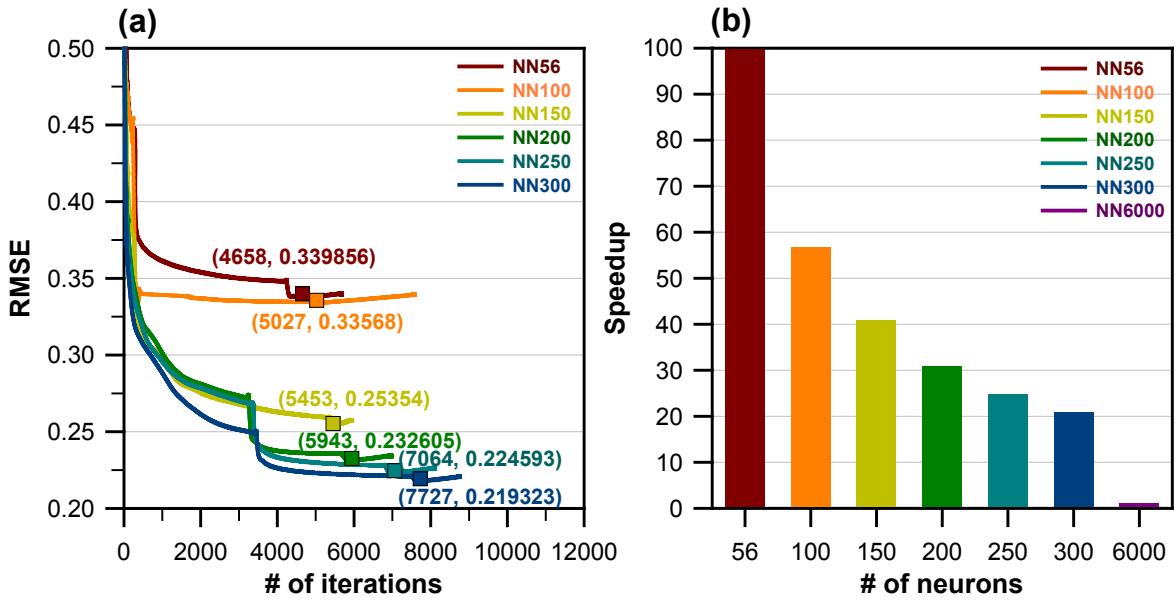


Figure S2. (a) Learning curves showing the root mean square error (RMSE) with iterations during the neural network (NN) training processes using different neurons (56–300) and (b) the comparison of speedup when emulators are based on coefficients with those neurons are applied to the WRF model. The use of 6000 neurons approximately corresponds to the case in which computational speedup by the emulator is impossible (i.e., the same computational cost to the RRTMG-K radiation parameterization).

Table S1. Lists of input and output variables.

Inputs	#
Pressure	1–39
Temperature	40–78
Water Vapor Mixing Ratio	79–117
Ozone Mixing Ratio	118–156
Cloud Fraction	157–192
Forecast Time	193
Surface Temperature	194
Solar Constant	195
Cosine Solar Zenith Angle	196
Outputs	#
Total Sky Longwave Radiative Heating Rate	1–39
Total Sky Shortwave Radiative Heating Rate	40–78
Total Sky Longwave Upward Flux at TOA (LWUPT)	79
Clear Sky Longwave Upward Flux at TOA (LWUPTC)	80
Total Sky Longwave Upward Flux at Bottom (LWUPB)	81
Clear Sky Longwave Upward Flux at Bottom (LWUPBC)	82
Total Sky Longwave Downward Flux at Bottom (LWDNB)	83
Clear Sky Longwave Downward Flux at Bottom (LWDNBC)	84
Total Sky Shortwave Upward Flux at TOA (SWUPT)	85
Total Sky Shortwave Upward Flux at Bottom (SWUPB)	86

Table S2. The root mean squared error (RMSE) and the square of the correlation coefficient (in parentheses) of predicted outputs for training and independent test sets with the number of hidden neurons.

# of neurons	LW heating rate	SW heating rate	LW flux	SW flux
Training sets				
56	1.114480 (0.881901)	0.732183 (0.816100)	2.322070 (0.999307)	24.404335 (0.987782)
100	1.317564 (0.900728)	0.769993 (0.817698)	2.390966 (0.999411)	22.629657 (0.993243)
150	1.162869 (0.924721)	0.575730 (0.899341)	1.885151 (0.999634)	18.230951 (0.995724)
200	0.794262 (0.937811)	0.417664 (0.939086)	1.315080 (0.999790)	13.932331 (0.995911)
250	0.773724 (0.941910)	0.406826 (0.941674)	1.232032 (1.000000)	12.765319 (0.996620)
300	1.020720 (0.941516)	0.490101 (0.926777)	1.576838 (0.999748)	14.393759 (0.997313)
Test sets				
56	1.598458 (0.871080)	1.152631 (0.729848)	3.104169 (0.999033)	33.879673 (0.988433)
100	1.544098 (0.881430)	1.132595 (0.741960)	2.582497 (0.999333)	25.778270 (0.993378)
150	1.349436 (0.908156)	0.931475 (0.823755)	1.985918 (0.999606)	21.247993 (0.995470)
200	1.120933 (0.928431)	0.548987 (0.907137)	1.838057 (0.999652)	18.599440 (0.995541)
250	1.173152 (0.931049)	0.809782 (0.867771)	1.687019 (0.999715)	19.070685 (0.996371)
300	1.211869 (0.926473)	0.847947 (0.855052)	1.656414 (0.999726)	18.279062 (0.996672)

Table S3. Same as Table 1, but for the use of different hidden neurons from 250 to 100. The statistics in the table are the root mean squared error (RMSE) and the square of the correlation coefficient (in parentheses) relative to the control run with the same computational cost.

Experiments	NN250	NN200	NN150	NN100
Speedup of radiation	24.77	30.76	40.80	56.67
Reduced computation time	83.31%	84.30%	85.03%	85.50%
LW heating rate [K day ⁻¹]	0.95 (0.92)	0.97 (0.91)	1.00 (0.91)	1.14 (0.88)
SW heating rate [K day ⁻¹]	0.48 (0.86)	0.49 (0.85)	0.47 (0.86)	0.48 (0.86)
Cloud fraction [%]	6.27 (0.97)	6.46 (0.97)	6.23 (0.97)	6.87 (0.96)
LW flux [W m ⁻²]	6.10 (1.00)	6.29 (1.00)	6.85 (1.00)	6.08 (1.00)
LWUPT	11.61 (0.90)	11.41 (0.90)	12.32 (0.88)	10.67 (0.92)
LWUPTC	2.06 (0.84)	2.31 (0.80)	2.40 (0.77)	2.03 (0.83)
LWUPB	1.11 (0.83)	1.20 (0.81)	1.24 (0.81)	1.14 (0.82)
LWUPBC	0.75 (0.96)	0.88 (0.95)	1.11 (0.93)	0.99 (0.93)
LWDNB	7.53 (0.82)	8.01 (0.79)	8.19 (0.80)	7.52 (0.81)
LWDNBT	5.10 (0.96)	5.96 (0.94)	7.33 (0.92)	6.72 (0.92)
SW flux [W m ⁻²]	43.23 (0.98)	45.54 (0.98)	43.28 (0.98)	44.29 (0.98)
SWUPT	59.81 (0.96)	62.92 (0.96)	59.84 (0.96)	61.36 (0.96)
SWUPB	12.65 (0.96)	13.73 (0.95)	12.86 (0.96)	12.59 (0.96)
Surface temperature (T_s) [K]	1.06 (0.92)	1.21 (0.91)	1.21 (0.91)	1.14 (0.90)
Precipitation [mm]	0.18 (0.61)	0.18 (0.58)	0.22 (0.42)	0.19 (0.55)

Table S4. Same as Table 1, but for different activation functions with 300 neurons. The statistics in the table are the root mean squared error (RMSE) and the square of the correlation coefficient (in parentheses) relative to the control run: 1) sigmoid ($x = 1/(1+e^{-x})$); 2) softsign ($x = x/(1+|x|)$); 3) arctan ($x = \tan^{-1}(x)$); 4) linear ($x = x$). Each of those functions has a different range such as $[0, 1]$, $[-1, 1]$, $[-\pi/2, \pi/2]$, and $[-\infty, \infty]$, respectively. Note that the range of tanh is the same as the softsign.

Experiments	sigmoid	softsign	arctan	linear
LW heating rate [K day^{-1}]	0.98 (0.91)	1.09 (0.89)	1.13 (0.88)	1.47 (0.81)
SW heating rate [K day^{-1}]	0.51 (0.85)	0.55 (0.82)	0.48 (0.86)	0.77 (0.68)
Cloud fraction [%]	6.71 (0.96)	7.13 (0.96)	6.79 (0.96)	7.74 (0.95)
LW flux [W m^{-2}]	6.47 (1.00)	6.44 (1.00)	6.89 (1.00)	8.49 (0.99)
LWUPT	12.23 (0.89)	11.75 (0.90)	12.80 (0.88)	17.07 (0.80)
LWUPTC	2.51 (0.76)	2.30 (0.81)	2.29 (0.79)	2.46 (0.77)
LWUPB	1.21 (0.81)	1.25 (0.79)	1.26 (0.78)	1.35 (0.75)
LWUPBC	0.82 (0.95)	0.96 (0.94)	0.99 (0.93)	1.10 (0.92)
LWDNB	8.00 (0.79)	8.14 (0.79)	8.37 (0.77)	8.95 (0.74)
LWDNBC	5.41 (0.95)	6.06 (0.94)	6.55 (0.93)	7.24 (0.91)
SW flux [W m^{-2}]	45.43 (0.98)	40.68 (0.99)	45.16 (0.98)	77.43 (0.95)
SWUPT	62.78 (0.96)	56.22 (0.97)	62.43 (0.96)	107.35 (0.88)
SWUPB	13.64 (0.95)	12.22 (0.96)	13.49 (0.95)	21.61 (0.88)
Surface temperature (T_s) [K]	1.19 (0.91)	1.39 (0.89)	1.24 (0.89)	1.43 (0.87)
Precipitation [mm]	0.19 (0.55)	0.22 (0.49)	0.21 (0.49)	0.21 (0.45)

Table S5. Same as Table 1, but for the evaluation with 56-neurons emulators from initial soundings with the cold-type and warm-type heavy rain perturbations. The cold-type and warm-type perturbations were adopted from Song and Sohn (2015) and Song et al. (2017). Thermodynamically more unstable conditions were prescribed to the cold-type experiment, whereas more stable but humid environments were considered the warm-type.

Experiments	NN56 (cold-type)	WRF100 (cold-type)	NN56 (warm-type)	WRF100 (warm-type)
LW heating rate [K day ⁻¹]	1.07 (0.90)	1.26 (0.87)	0.88 (0.93)	0.90 (0.93)
SW heating rate [K day ⁻¹]	0.50 (0.85)	0.55 (0.83)	0.40 (0.88)	0.44 (0.88)
Cloud fraction [%]	6.26 (0.97)	7.43 (0.96)	5.22 (0.98)	9.07 (0.94)
LW flux [W m ⁻²]	7.71 (0.99)	8.63 (0.99)	5.70 (1.00)	7.36 (0.99)
LWUPT	14.86 (0.86)	17.31 (0.81)	12.08 (0.87)	15.28 (0.80)
LWUPTC	3.32 (0.73)	3.02 (0.78)	1.81 (0.77)	2.19 (0.70)
LWUPB	1.41 (0.74)	1.38 (0.76)	0.90 (0.85)	1.15 (0.77)
LWUPBC	1.11 (0.92)	1.07 (0.93)	0.72 (0.95)	0.78 (0.95)
LWDNB	8.66 (0.76)	9.19 (0.74)	5.36 (0.85)	7.64 (0.74)
LWDNBT	6.85 (0.92)	7.15 (0.92)	4.00 (0.96)	5.18 (0.95)
SW flux [W m ⁻²]	55.74 (0.97)	65.67 (0.96)	43.11 (0.98)	57.20 (0.97)
SWUPT	77.00 (0.94)	90.45 (0.91)	59.47 (0.96)	78.71 (0.93)
SWUPB	16.89 (0.92)	21.03 (0.88)	13.45 (0.95)	18.66 (0.91)
Surface temperature (T _s) [K]	1.36 (0.86)	1.69 (0.82)	0.74 (0.94)	1.10 (0.91)
Precipitation [mm]	0.20 (0.44)	0.20 (0.47)	0.15 (0.74)	0.17 (0.67)