Optimization Learning Report

Skyler Ebelt
Georgia Institute of Technology
GT ID: 904077010

I. AI Use Statement

In order to complete this project, AI was utilized in a few ways. I only used Claude Sonnet 4 for the duration of this project. Claude was utilized for organization, to better manage my workflow on this project. Additionally, Claude was used to refactor broken code segments. I worked to understand every change suggested by Claude before implementing it in my code.

II. Introduction

A. Hypothesis

This analysis seeks to examine, measure, and analyze the impacts of optimization on neural network model performance. The use of optimization should significantly improve model performance, providing better generalization with more tools utilized to better capture trends in the provided datasets. In other words, the models presented in this project are expected to outperform the baseline neural network models taken from project one.

B. Datasets Utilized

In order to complete this project, hotel booking data [1] and US accidents data [2] was once again utilized. These datasets serve different purposes entirely, and greatly vary in dimensionality, length of observations, and capture much different trends. Because of this, these work well to capture the effects of optimization, be it a performance improvement or degradation. With a foundational understanding of this data after the thorough analysis from project one, this analysis will be able to accurately measure and describe how optimization effects these models, where the algorithms work well, and where they can fall short. On a personal sidenote, I simply find classification problems much more interesting which is why I have opted to address these in both projects to date. Perhaps I will change it up and take the regression route soon, although that simply sounds like less fun to me. From a real world standpoint as well, classifying the target variables in datasets seems to me to have a more impactful real world application. With being able to predict cancellations in the case of hotel data, this can help hotels run much more efficiently and retain clientele. Whereas in the case of the accidents data, predicting severity and implementing such models on a large scale can save lives. It is for these reasons that these datasets have been utilized in this manner.

C. Objectives

The objective of this project was to implement different optimization algorithms to measure the impact on performance within neural network models trained on the datasets outlined above. This performance has been compared to the previous performance of NN models from project one, that being the analysis of different models performances trained on the same data. The best NN models from project one were implemented initially to provide a baseline performance of NN models without any optimization techniques implemented.

III. Data & Methodology

A. Hotel Data

Hotel data features approx. 119,000 observations with 32 features. I once again opted to address classification for both datasets meaning that once again the columns "reservation_status", "reservation_status_date" were dropped to prevent leakage. Models were stratified on/the target variable was "is_canceled". The same preprocessing was used from project one for this project which involved converting float64 into float32, as well as handling high cardinality columns by dropping them. Once again, in the interest of computational feasibilty, a tight threshold of 20 unique variables per feature was selected, should a feature exceed this limit, it be encoded using target encoding rather than one hot encoding.

B. Accidents Data

Accidents data similarly was modified in the same way as project one due to the pursuit of addressing the classification element of this data. Post outcome data was removed which were composed of the features "end_time" and "weather_timestamp". This data boasts > 7,000,000 observations with 46 features. High cardinality was handled in the same way as hotel data, > 20 unique variables resulted in target encoding for the large cardinality columns. Float64 variables were handled by converting them to float32. Both datasets were preprocessed in the same manner as project 1, to try to ensure same to similar performance for an accurate baseline to compare with.

C. Evaluation & Hardware

After struggling with runtime in project one, and learning a great deal from that, I opted to swap from CPU torch vectors to CUDA. Armed with 32 GB of DDR4 ram, and intel i7 9700 @ 3.00 GHZ and a NVIDIA RTX 2080 super, I am heavily bottle necked by my CPU, frequently maxing it out during runtime. Having access to a solid GPU meant that swapping to CUDA vectors seemed the most sensible thing. Because of this, this analysis was able to run smoothly despite heavy CPU sided bottlenecks. Loss was the main metric being evaluated across different optimization algorithms. This included both training and validation loss, occasional other metrics were calculated and implemented in the analysis such as generalization gaps. Seeds and random states were associated with my GTID (4077010) and for testing multiple seeds I simply appended the iteration of the seed to the end of GTID (EX: 40770100, 40770101,...,40770104). Unlike project one, changes to core functionality or sampling was not required with the exception of Adam ablation on accidents data, some runtimes were hefty, however they were not stalling or crashing nearly as frequently. Only one example of a stall tends to occur in genetic algorithms, with both algos on each dataset stalling in the 450-499 range. To fix this I simply interrupted the kernel and proceeded with the analysis in the case of hotels. In the case of accidents, I limited the max evals to 400. An example of sampling can be found within adam ablation on accidents data, this was taking upwards of 8 hours to run prior to sampling, this simply was not feasible for me.

IV. Randomized Optimization

A. Implementation

In order to experiment with randomized optimization, three algorithms were implemented on neural network architectures. These architectures remained consistent with that of the architectures present in the supervised learning project. This allows for a better comparative analysis of neural networks pre and post optimization to measure change in performance, dataset specific limitations and trends, etc. Both models featured nine layers, with the final three being selected for optimization, the first six layers were frozen to remain consistent across experimentation. Within these unfrozen layers, hotel models had 11,577 trainable parameters whereas accident models featured 11,679 trainable parameters. This conforms to the project guidelines of ensuring models are training less than or equal to 50,000 parameters.

The three algorithms were established with identical evaluation budgets of 500, with the exception of genetic algorithms. Due to stalling in the final evaluations of both hotel and accident models, evaluations were cut to 400 in the accident model (with stalling occurring around the 450 evaluation mark) and with keyboard interrupts to prevent stalling in the hotel model (usually occuring between 450-499) evaluations. Randomized Hill Climbing utilizes a greedy search technique with restarts every 100 evaluations and exponential step size decay (factor 0.95). The initial perturbation scale of 0.01. Simulated Annealing used a Metropolis criterion with exponential temperature cooling from an initial value of 1.0. The Genetic Algorithm maintained a population of 20 individuals with fitness-proportional selection, uniform crossover for offspring generation, mutation rate of 0.1, and elitism to preserve the best solution across generations.

B. Results & Analysis

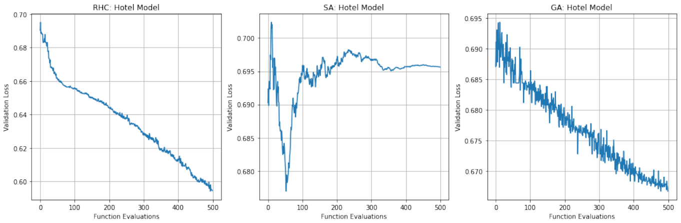

Fig. 1. Loss Functions of RO on Hotel Data

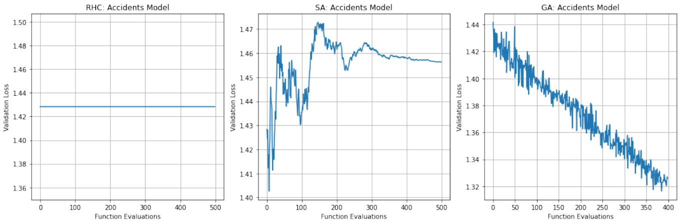

Fig. 2. Loss Functions of RO on Accidents Data

TABLE I
Randomized Optimization Performance Comparison

Dataset	Algorithm	Baseline Loss	Final Loss	Improvement
Accidents	RHC	1.4284	1.4284	0.0000
Accidents	SA	1.4284	1.4022	0.0262
Accidents	GA	1.4137	1.3443	0.0694
Hotel	RHC	0.6903	0.5988	0.0915
Hotel	SA	0.6903	0.6570	0.0333
Hotel	GA	0.6864	0.6549	0.0315

The three algorithms used in this analysis showed varying performance when comparing performance two datasets, highlighting trends or noise specific to each dataset. In the hotel dataset, random hill climbing performed best, achieving a 13.26% improvement in validation loss. This allowed hill climbing to efficiently descend to a local optimum.The periodic restarts provided additional exploration opportunities. The model appeared to quickly converge, requiring few restarts to find the best loss.

Randomized hill climbing completely failed on the accident data set, maintaining the baseline loss of 1.4284. This would indicate that every attempted perturbation was rejected by the acceptance criterion. This behavior suggests perhaps that the initialization resided in an extremely flat region where small perturbations produced negligible loss changes. The chosen initial scale of 0.01, worked well in the case of the hotel model but failed on the accidents model.

Simulated annealing demonstrated intermediate performance in both datasets, achieving a 1.83% improvement in accident data and a 4.82% improvement in hotel data. The validation loss can be seen fluctuating between 1.42 and 1.45 during the first 100 iterations. The exploratory behavior of this model allowed SA to escape the local region that trapped RHC, gradually converging as the temperature cooled. The

Hotel trajectory settles similarly into a descent pattern around iteration 200.

The Genetic Algorithm produced the best Accidents performance with 4.91% improvement, exhibiting steady descent with occasional discrete jumps. In Hotel, GA achieved 4.59% improvement, performing comparable to SA but not as well as RHC.

The hotel dataset appears to possess relatively smooth local geometry, favoring hill climbing. The frozen early layers, trained on hotel booking patterns, seem to represent features enabling this models performance increase over our baseline. The Accidents dataset presents a more challenging optimization landscape, which defeated the perturbations employed by RHC. The complete failure of RHC on Accidents, despite success on Hotel under identical hyperparameter settings, underscores the importance of algorithm selection due to conditions unique to any given dataset.

In the case of accident data, GA did will to optimize this algorithm, without falling short as can be seen in hill climbing. This indicates that population-based exploration was perhaps better at navigating a landscape which was far too complex for RHC. This might suggest many points of local minima/maxima, which may explain why RHC failed with this model.

These results demonstrate that gradient-free optimization may be useful in boosting model performance. Though, model selection and development must be carefully planned, implemented, and executed to ensure that optimization methods selected are well suited for the landscape provided by the data being utilized for modeling. RHC in the case of hotel data, did well to improve model performance, as shown by the significant increase in the loss function, similarly we can observe similar results within GA on accidents data, boasting an even higher improvement.

V. Adam Ablation

A. Implementation

This study required implementing and evaluating seven different optimization algorithms on entire neural network architectures. This differs from the initial study of this paper which only optimized on the final 3 layers of the neural network architecture. The impact of momentum, learning rates, weight decay, etc. was analyzed and measured to evaluate optimizer specific effects, spanning the aforementioned 2 separate data sets (hotel data and accident data) which also provides a view into data specific effects.

Seven distinct optimizer algorithms were implemented to measure the effects of each one respectively. The baseline, which was established for comparison, included three SGD variants: stochastic gradient descent without momentum, SGD with momentum, and SGD with Nesterov accelerated gradients. A learning rate of 0.01 was used in pursuit of establishing a benchmark. Variants of Adam utilized were comprised of, standard Adam with bias correction, Adam with B1 = 0 to emulate RMSProp behavior, and AdamW with decoupled weight decay. All Adam variants utilized a learning rate of 0.001. For training, models were allowed 15 epochs with batch sizes of 512. L2 weight decay regularization of 1e-4 was applied. Five independent training runs occurred using each model using seeds which followed the previously used convention, for example: 40770100, 40770101…40770104.

B. Results & Analysis

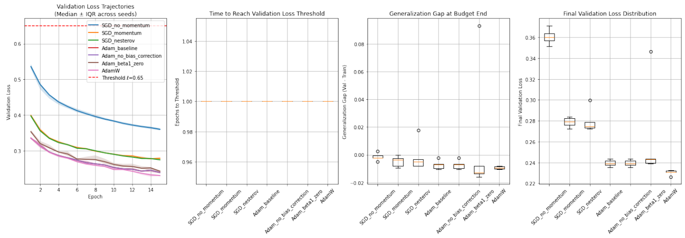

Fig. 3. Hotel Data: Adam Ablation Performance Plots

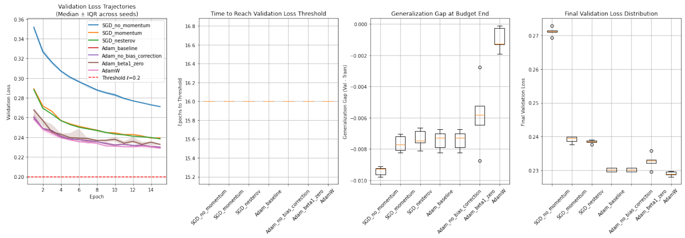

Fig. 4. Accidents Data: Adam Ablation Performance Plots

TABLE II
Adam Ablation Study: Accidents Dataset Performance

Optimizer	**Val Loss**	**Gen Gap**	**IQR**
SGD (no momentum)	0.2711 ± 0.0004	-0.0093	0.0004
SGD + Momentum	0.2394 ± 0.0013	-0.0077	0.0013
SGD Nesterov	0.2384 ± 0.0003	-0.0075	0.0003
Adam (baseline)	0.2301 ± 0.0011	-0.0073	0.0011
Adam (no bias corr.)	0.2301 ± 0.0011	-0.0073	0.0011
Adam ($\beta_1 = 0$)	0.2328 ± 0.0011	-0.0058	0.0011
AdamW	$\mathbf{0.2288 \pm 0.0010}$	$\mathbf{-0.0013}$	0.0010

TABLE III
Adam Ablation Study: Hotel Dataset Performance

Optimizer	**Val Loss**	**Gen Gap**	**IQR**
SGD (no momentum)	0.3602 ± 0.0069	-0.0022	0.0069
SGD + Momentum	0.2790 ± 0.0064	-0.0034	0.0064
SGD Nesterov	0.2748 ± 0.0054	-0.0051	0.0054
Adam (baseline)	$\mathbf{0.2390 \pm 0.0039}$	-0.0070	0.0039
Adam (no bias corr.)	0.2390 ± 0.0039	-0.0070	0.0039
Adam ($\beta_1 = 0$)	0.2431 ± 0.0041	-0.0128	0.0041
AdamW	0.2385 ± 0.0014	$\mathbf{-0.0093}$	0.0014

This study required implementing and evaluating seven different optimization algorithms using entire neural network architectures. This differs from the initial study of this paper which was only optimized on the final 3 layers of the neural network architecture. The impact of momentum, learning rates, weight decay, etc. was analyzed and measured to evaluate optimizer specific effects, spanning the aforementioned two separate models trained on different data sets (hotel data and accident data) which also provides a view into data specific effects.

Seven different optimizer algorithms were implemented to measure the effects of each, respectively. The baseline, which was established for comparison, included three variants of SGD: stochastic gradient descent without momentum, SGD with momentum, and SGD with Nesterov accelerated gradients. A learning rate of 0.01 was used to establish a benchmark. The Adam variants utilized were composed of standard Adam with bias correction, Adam with B1=0 to emulate RMSProp behavior, and AdamW with decoupled weight decay. All Adam variants utilized a learning rate of 0.001. For training, models were allowed 15 epochs with batch sizes of 512. L2 weight decay regularization of 1e-4 was applied. Five training runs occurred using each model using seeds that followed the previously used convention, for example: 40770100, 40770101…40770104.

Establishing the optimization algorithms, allowed for the observation of different patterns/behaviors across each instance. This provided a view of sensitivities between optimizer choice and hyperparameters. Within accidents, AdamW achieved a median loss validation of 0.2288, performing better than the other instances of Adam. Standard Adam reached 0.2301, and Adam with B1=0 at 0.2328. Among SGD variants, Nesterov momentum outperformed standard momentum, meanwhile vanilla SGD without momentum had the highest loss of 0.2711. When looking at the convergence it reveals behavioral patterns between optimizers. The SGD without momentum exhibited extremely slow convergence, requiring all 15 epochs to fully converge, while failing to reach the validation loss threshold of 0.2, suggesting more epochs may have been necessary. The addition of momentum accelerated convergence, with both standard and Nesterov variants showing a quick descent followed by refinement. AdamW's strong final performance suggests that decoupled weight decay provides more effective regularization than L2 penalty.

When considering the hotel data and the optimization of its models better results were netted by the techniques utilized, with all algorithms converging within a few epochs and reaching the validation threshold of 0.65 within a single epoch. Standard Adam achieved the best median validation loss of 0.2390, while AdamW reached a comparable 0.2385. This contrasts the Accidents dataset, suggesting that Hotel's classification task involves more separable decision boundaries that accommodate various optimization strategies. The Nesterov momentum outperformed the standard momentum. Finally the Vanilla SGD was slghtly ahead of AdamW reaching a score of 0.3602.

All configurations showed negative generalization gaps, which is a sign of underfitting in these instances. In the Accidents dataset, generalization gaps ranged from -0.0013 (AdamW) to -0.0093 (vanilla SGD), with adaptive methods generally showing smaller absolute gaps. AdamW's minimal gap of -0.0013 suggests a good balance between training performance and validation generalization, while the vanilla SGD's larger gap of -0.0093 indicates some remaining capacity to reduce training loss. The hotel model showed similar gaps from -0.0022 to -0.0128, while Adam B1=0 having the largest gap of -0.0128.

Using five different seeds to perform a stability analysis showed similar behavior for all optimizers, with the IQR remaining tight. In accidents, IQR values ranged from 0.0003 to 0.0013. The Hotel data set exhibited slightly larger but still smaller IQR values from 0.0014 to 0.0069, indicating that optimizer performance is not heavily dependent on random initialization and that the observed performance differences outline behavior specific to each algorithm.

The varying optimization challenges between the Accidents and Hotel datasets shed light on how the features of the task affect both the choice and effectiveness of the optimizer. The relatively quick convergence of the Hotel dataset within a single epoch for all optimizers indicates a relatively simple loss landscape. The task of binary classification for predicting hotel booking cancellations is likely to have linearly separable decision boundaries. The much slower convergence of the Accidents dataset and its greater sensitivity to the choice of optimizers, may suggest a more difficult optimization landscape, marked by weak gradients, many local minima, or intricate interactions between input features. Research into the Adam optimization method may provide more insight into why the accidents model seemed to present signs of difficulty in increasing model performance. Difficulty to properly converge may be the result of skewedness within the gradients when optimizing the accidents model [3]. These results appear to be consistent with the research found in *Theoretical Analysis of Adam Optimizer in the Presence of Gradient Skewness* which describes phenomena such as slow or poor convergence which arises in the case of skewed gradients. When considering this research, methods such as distribution aware optimization algorithms may have remedied this issue. It is also entirely possible, this example of a deep NN model may be achieving near perfect performance as per its architecture. In other words, these methods may not improve performance very much, as it is limited by its architecture (although it is by no means performing poorly).

VI. Normalization

A. Implementation

The study on regularization explored five methods working within the standard Adam optimizer, using hyperparameters identified as having provided the best results from the analysis within Part 2. Unlike Part 2, which uniformly applying weight decay to all optimizers, the baseline setup utilized pure Adam optimization which did not incorporate any weight decay. The five regularization techniques were tested under the same training conditions, 15 epochs, a batch size of 512, and five different seeds. I used the same seeds as before which were outlined in previous sections. Early stopping was used, monitoring validation loss with a patience of 3 epochs and a minimum improvement threshold of 10≙4, preventing training when validation performance was found to no longer increase. A dropout rate of 0.3 was applied between hidden layers. Label smoothing adjusted hard classification targets into distributions, using a smoothing parameter of alpha=0.1, which replaced one-hot labels y with smoothed targets. During training, augmentation was implemented by adding Gaussian noise with a standard deviation of 0.01 to input features.

B. Results & Analysis

TABLE IV
REGULARIZATION STUDY: ACCIDENTS DATASET PERFORMANCE

Method	Test Accuracy	vs Baseline	IQR
Baseline	0.9181 ± 0.0003	-	0.0003
L2 Weight Decay	0.9171 ± 0.0001	−0.10%	0.0001
Early Stopping	0.9188 ± 0.0009	+0.07%	0.0009
Dropout	0.9167 ± 0.0005	−0.14%	0.0005
Label Smoothing	**0.9182 ± 0.0007**	+0.01%	0.0007
Augmentation	0.9183 ± 0.0009	+0.02%	0.0009

TABLE V
REGULARIZATION STUDY: HOTEL DATASET PERFORMANCE

Method	Test Accuracy	vs Baseline	IQR
Baseline	0.8701 ± 0.0012	-	0.0012
L2 Weight Decay	0.8696 ± 0.0033	−0.05%	0.0033
Early Stopping	0.8715 ± 0.0015	+0.14%	0.0015
Dropout	0.8724 ± 0.0016	+0.23%	0.0016
Label Smoothing	**0.8737 ± 0.0023**	+0.36%	0.0023
Augmentation	0.8700 ± 0.0050	−0.01%	0.0050

The regularization techniques once again showcased how understanding the data provided in a task is crucial to optimization. In the accidents dataset, all techniques showed minor performance shifts compared to the baseline models, with test accuracies ranging from 0.9167 to 0.9183.Augmentation achieved the highest test accuracy of 0.9183, while label smoothing nearly reflected this performance at 0.9182. Early stopping produced a small improvement to 0.9188. Two techniques actually degraded accident performance below baseline, dropout reduced accuracy to 0.9167, and L2 weight decay decreased performance to 0.9171.

The hotel dataset demonstrated greater sensitivity to regularization, with methods showing a 0.4% range in performance and many achieving significant improvements over the initial baseline. Label smoothing was the highest performer a test accuracy of 0.8737, yielding a 0.36% increase over the baseline of 0.8701. Once again an example of how the same methods work much better when the models being optimized are trained on the hotel data. This more substantial effect size for Hotel, compared to accidents, implies that the Hotel classification task has a higher risk of model overfitting. Dropout achieved the second-highest performance with 0.8724, suggesting that the hotel model has sufficient capacity to benefit from enforced redundancy. Augmentation proved to have little effect on hotel data, and L2 weight decay again showed a slight decrease in the efficacy of the model.

The minimal improvements from regularization on Accidents, combined with negative generalization as seen in part 2, reveal that the baseline model exhibits slight underfitting. These regularization techniques seem to fail to help the model further generalize, this may be due to the regularization making certain patterns within the data unreadable to the model, not adding enough context to actually help the models decision making, or it the model simply may not possess the ability to improve. A similar pattern can be found here as in part 2. Model performance is decently high, but not increasing as might be expected from using these techniques. Because regularization is favorable in reducing overfitting [4], if a model is not already prone to overfitting this technique may not support its performance. Because of this, it would seem that despite performance not increasing, this model is performing well and may indicate a well generalized model.

The Hotel data set's greater responsiveness to regularization indicates a model closer to the overfitting boundary, providing improvements to the efficacy of the model. Label smoothing with alpha=0.1 transforms one hot target into distributions. This seems to encourage the model to maintain uncertainty in its predictions and reduce the penalty for close to correct predictions. Dropout's 0.23% improvement on the Hotel dataset, contrasted with a 0.14% decrease on the Accidents dataset, highlights how this method's effectiveness depends on capacity. It reduces model capacity by deactivating 30% of neurons at random while in the training phase, essentially forcing the network to learn backup representations. The Accidents task achieves a 91.81% baseline accuracy, indicating that the classes are either easily separated or the prediction error is high due to uncertainty. The negative generalization gap from Part 2 shows that the model doesn't fit the training data well. In this scenario, using regularization techniques to limit capacity won't enhance performance because the model is already capacity-constrained. The Hotel dataset's 87.01% baseline accuracy and higher sensitivity to regularization imply a task with more intricate decision boundaries, increasing the risk of overfitting.

VII. CONCLUSION

This analysis examined whether optimization techniques would improve neural network performance beyond the established baselines. The results provide mixed support for this hypothesis.

Part 1's randomized optimization experiments show limited value. In accidents data, the best method achieved only slight improvement, while Randomized Hill Climbing completely failed. RHC improved by 13% in hotel data, but gradient-free methods are still impractical, needing over 10,000 function evaluations for networks with thousands of parameters for any significant improvements.

Part 2's Adam ablation study confirmed that choosing the right optimizer significantly affects performance. AdamW outperformed vanilla SGD by 15.6% in accidents data, and Adam beat SGD by 33.6% in hotel data. The choice between first-order and adaptive methods matters much more than any other optimization decision. However, the performance differences between the Adam variants remained small.

Part 3's regularization study produced no results in support of the hypothesis. In accidents data, regularization provided nearly no benefit, with the best technique improving accuracy by only 0.02% while dropout actually affected performance by 0.14%. Hotel data showed greater responsiveness with label smoothing, gaining 0.36%. The baseline models already achieved strong generalization, leaving little room for regularization to improve.

The two datasets revealed fundamentally different optimization landscapes. Accident data proved challenging for randomized methods but easy for gradient-based training, achieving 91.8% accuracy with little help from regularization techniques. The hotel data presented an easier landscape overall, with all methods successful, but more room for improvement in regularization at 87.0% of the baseline precision.

These patterns outline the shortcomings of the initial hypothesis, the techniques utilized vary greatly between the two datasets. This shows how decisions about the selection of optimization techniques are much more complex. The landscape presented by the data and the trends or patterns they showcase can defeat some algorithms but can be greatly supported by others. It is not true across the board that optimization will help any given models performance, in many cases model performance can be degraded as shown in some of these analyses. Some analyses within the accidents model even suggest optimization is not required/is helping but not to a meaningful degree. This may suggest changing model architecture could be just as effective as any given optimization technique. Because of this, thorough testing must be performed to ensure that a practitioner selects the right optimization techniques, ensuring they are helpful and worth the investment of time.

REFERENCES

[1] Hotel Booking Demand: Data: Hotel booking demand (H1/H2), Ant´onio, Almeida & Nunes (2019), Data in Brief 22:41–49, doi:10.1016/j.dib.2018.11.126.

[2] US Accidents: Data: US Accidents (since 2016), Sobhan Moosavi et al., CC BY-NC-SA 4.0, retrieved from Kaggle US Accidents.

[3] Theoretical Analysis of Adam Optimizer in the Presence of Gradient Skewness. Yang, Luyi. (2024).International Journal of Applied Science. 7. p27. 10.30560/ijas.v7n2p27.

[4] Empirical Evaluation of the Effect of Optimization and Regularization Techniques on the Generalization Performance of Deep Convolutional Neural Network. Marin, I., Kuzmanic Skelin, A., & Grujic, T. (2020). Applied Sciences, 10(21), 7817. https://doi.org/10.3390/app10217817