Exploring the Impact of Noise on Neural Networks: A Computational Perspective

By:

Odunayo Adekoya

(21053999, SYDE MEng)

1. Abstract

Understanding the impact of noise on neural network generalization is crucial for improving model robustness and real-world applicability.

This paper investigates the influence of various types of noise, including Gaussian, dropout, and adversarial noise, on the generalization performance of neural networks in computational simulations. We aim to discern how these forms of noise affect model behavior and to what extent these findings extrapolate to real-world neural systems.

We conducted a series of experiments wherein neural network architectures were trained and evaluated under different noise conditions. Gaussian noise, dropout noise, and adversarial noise were systematically introduced during training and testing phases. Model performance metrics, including accuracy and loss, were measured across multiple trials to assess generalization capabilities.

Our experiments revealed nuanced effects of different types of noise on neural network generalization. While higher levels of gaussian noise acted as a disruptor, dropout noise demonstrated imbalance over time. However, adversarial noise had a negligible impact.

The findings underscore the importance of considering noise in neural network training and testing processes. Future research should focus on developing robust techniques to mitigate the impact of gaussian and dropout noise and further explore the transferability of these findings to real-world neural systems.

2. Introduction

Understanding the robustness and generalization capabilities of neural networks in the presence of noise is fundamental for advancing both theoretical understanding and practical applications in artificial intelligence. Noise, defined as any undesired or random perturbation affecting the input or internal states of a neural network, poses significant challenges to the reliable operation of these systems. In this paper, we aim to investigate how different types of noise influence the generalization performance of neural networks in computational simulations and explore the extent to which these findings transfer to real-world neural systems.

2.1 Problem Statement and Justification

The ability of neural networks to generalize well to unseen data is crucial for their effectiveness in real-world applications such as image recognition, natural language processing, and autonomous decision-making. However, the presence of various sources of noise during training and inference can undermine this generalization performance, leading to overfitting, decreased robustness, and susceptibility to adversarial attacks.

In this context, this research seeks to address the pressing need for a comprehensive understanding of how different types of noise influence the generalization performance of neural networks. By systematically investigating the effects of noise across various neural network architectures and tasks, we aim to uncover novel insights that can inform the development of robust and resilient AI systems.

2.2 Literature Review

Previous research has provided valuable insights into the effects of noise on neural network behavior. Zhang et al. (2019) demonstrated that the presence of Gaussian noise during training can act as a regularizer, improving generalization performance by reducing overfitting to the training data. Similarly, Srivastava et al. (2014) introduced dropout regularization as a technique to prevent co-adaptation of neurons in deep neural networks, leading to improved generalization on various image classification tasks.

Conversely, recent studies have highlighted the vulnerability of neural networks to adversarial attacks, which exploit small perturbations in input data to induce misclassification (Goodfellow et al., 2015). Adversarial noise, introduced during either training or testing, can severely degrade generalization performance and compromise the reliability of neural network predictions (Szegedy et al., 2013).

In addition to these well-studied forms of noise, recent research has also explored the effects of other types of perturbations on neural network behavior. For instance, Xie et al. (2020) investigated the impact of label noise on the generalization capabilities of neural networks, demonstrating that noisy labels can significantly degrade model performance and impede learning convergence.

This paper builds upon this existing literature by systematically investigating the influence of specific types of noise, including Gaussian noise, dropout noise, and adversarial noise, on generalization behavior across a range of neural network architectures and tasks. By synthesizing findings from previous studies and extending them through empirical experimentation, we aim to deepen our understanding of the complex interactions between noise, learning dynamics, and generalization behavior in neural networks.

2.3 Approach and Objectives

To rigorously investigate the influence of different types of noise on neural networks, we will formulate our experimental approach around a feedforward neural network as follows:

Model Formulation:

Considering a feedforward neural network with L layers, denoted as {W^(l), b^(l)}_{l=1}^{L}, where W^(l) represents the weight matrix and b^(l) is the bias vector for layer l. The activation function for hidden layers will be denoted as σ(.), and the softmax function will be used for the output layer to obtain class probabilities.

The output of the l-th layer, denoted as Z^(l), is computed as:

$$Z^{(l)} = \sigma(Z^{(l-1)} * W^{(l)} + b^{(l)})$$

where Z^(0) = X represents the input data matrix. The final output of the network, Y, is obtained by applying the softmax function to the output of the last layer:

$$Y = \text{softmax}(Z^{(L)} * W^{(L+1)} + b^{(L+1)})$$

Noise Formulation:

We will consider three types of noise commonly encountered in neural network training and inference:

Gaussian Noise: Denoted as "N(0, σ^2)", represents a random variable sampled from a Gaussian (or normal) distribution with zero mean (μ = 0) and variance σ^2. In the context of neural networks, Gaussian noise is often added to input data or internal representations to introduce variability and perturbations, simulating real-world noise sources.

The process of adding Gaussian noise to input data X can be mathematically represented as follows:

$X_{noisy} = X + N(0, \sigma^2)$

where:

X_{noisy} represents the noisy version of the input data X,

$N(0, \sigma^2)$ denotes the Gaussian noise with zero mean (μ = 0) and standard deviation σ.

In this equation, each element of the input data X is independently perturbed by a random sample drawn from the Gaussian distribution with mean zero and standard deviation σ. This introduces variability into the data, mimicking the effects of real-world noise sources such as sensor inaccuracies or measurement errors.

Dropout Noise: Dropout is a regularization technique commonly used in neural networks during training to prevent overfitting. It works by randomly setting a fraction of input units to zero at each update during training time, which helps prevent complex co-adaptations of neurons. Dropout noise can be represented as follows: Let x be the input to a layer in a neural network, and let p be the probability of retaining a unit (often set to 0.5). Then, the dropout noise x is computed as:

$$x = x/p \cdot \text{Bernoulli}(p).$$

Here, Bernoulli(p) represents a binary random variable that takes the value 1 with probability p and 0 with probability $1-p$.

During training, we will apply dropout regularization to randomly deactivate a fraction p of neurons in each hidden layer, effectively introducing noise into the network's activations.

Adversarial Noise: Adversarial noise refers to small, carefully crafted perturbations added to input data with the intent of misleading a neural network into making incorrect predictions. Mathematically, the adversarial noise η added to input data x can be represented as follows:

$$x(adv) = x + \eta.$$

The goal of generating adversarial examples is to find the perturbation η that maximizes the loss function with respect to the original input x, subject to some constraints. Perturbations will be added to input samples to craft adversarial examples X_adv such that the neural network misclassifies them. We will employ techniques like Projected Gradient Descent (PGD) to generate these adversarial perturbations.

Experimental Objectives:

Quantify Generalization Performance: To evaluate the performance of the feedforward neural network model trained with and without noise on held-out test data. Metrics such as accuracy, signal-to-noise ratio and loss are measured to quantify the model's behavior under different noise conditions.

Assess Robustness to Adversarial Attacks: Investigate the robustness of the trained model against adversarial attacks by measuring the success rate of attacks on the test data.

Analyze Transferability to Real-world Systems: Compare the findings from our computational simulations with reviewed literature. We will also examine whether the insights gained from the simulated experiments can be generalized to real-world scenarios, considering factors such as network architecture complexity and dataset characteristics.

3. Methods

3.1 Model Description

A feedforward neural network model is used for this experimentation, comprising multiple layers of interconnected neurons, organized in a sequential manner. Each neuron receives inputs from the previous layer, performs a weighted sum of these inputs, applies an activation function, and passes the result to the next layer. The model is compiled with the Adam optimizer, sparse categorical cross-entropy loss function, and accuracy metric. The model is trained using the fit method with the training data for between 10 & 20 epochs. Validation data is provided to monitor performance on unseen data during training. This model serves as our baseline for comparison.

3.2 Noise Introduction

Our intervention involves introducing various types of noise to the model during training and/or testing phases. The types of noise considered include:

1. **Gaussian Noise:** Gaussian noise is added to input data or neuron parameters. This kind of noise is characterized by a normal distribution with zero mean and a specified standard deviation(s).

2. **Dropout Noise:** Dropout noise is implemented during the training phase of the neural network. It randomly deactivates a fraction of neurons in each layer to prevent overfitting and encourage the network to learn more robust features.

3. **Adversarial Noise:** Adversarial noise is introduced to the input data or network parameters with the aim of perturbing the model's behavior. This noise is designed to deceive the model's predictions or induce misclassification. Projected Gradient Descent (PGD) was used to generate these adversarial perturbations in this experiment.

3.3 Simulation Setup

We aim to test the influence of different types of noise on the generalization performance of the neural network. To achieve this, we conduct multiple experiments where each type of noise is systematically introduced into the model under controlled conditions. We evaluate the model's performance on a separate validation dataset to assess its ability to generalize unseen data accurately. By comparing the performance metrics across experiments with and without noise interventions, we can infer the impact of each type of noise on the model's generalization capability.

3.4 Data Analysis

After simulations for over 10 iterations, key performance metrics such as accuracy, loss, and signal-to-noise ratio (SNR) were analyzed. In addition, data for the experiments with noise and without noise were compared. The recorded data, in the form of plotted graphs, were then collected and compared to the initial sinusoidal oscillatory graph to observe the effects of noise on the model's behavior. These analyses helped to gain insights into the models' predictive accuracy and robustness under different levels and types of noise.

4. Results

4.1 Initial Simulation

The initial results of the simulations shows a feedforward neural network model on an MNIST dataset without any noise over 10 epochs.

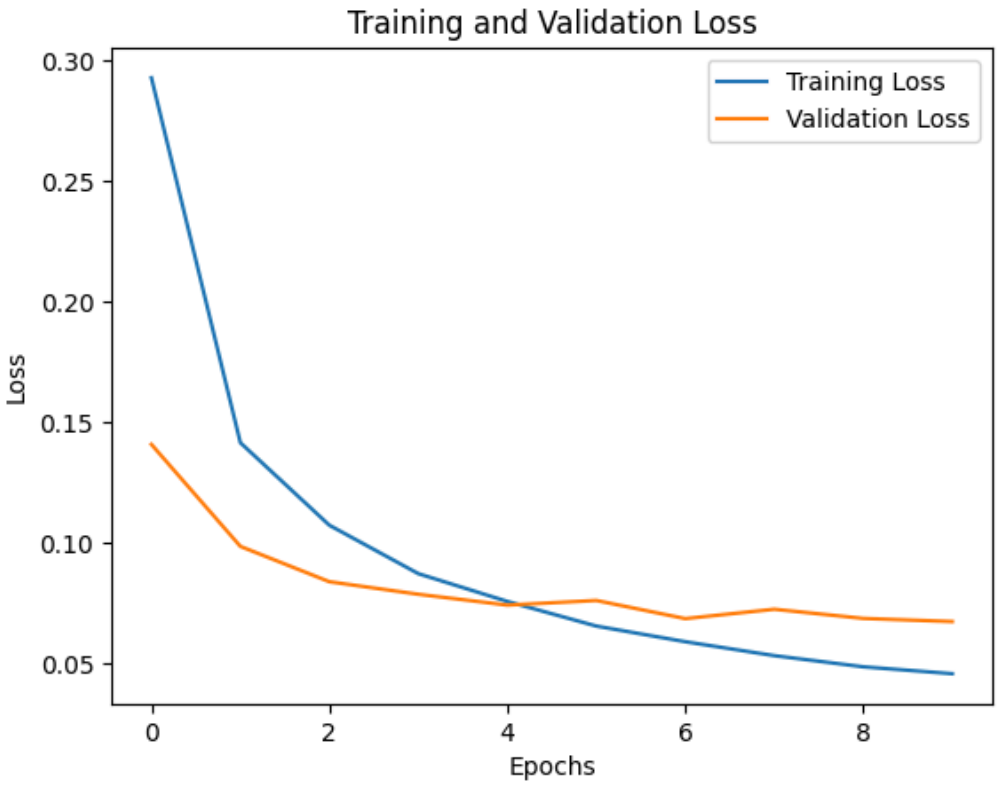

Figure 1: Training and validation loss curves plotted across epochs during the training of a neural network model on the MNIST dataset

4.2 Effect of Gaussian Noise on Input Layer

After introducing gaussian noise to the input signals fed into the feedforward neural network, the following plots depicts the effects of random noise on the model trained over 15 epochs:

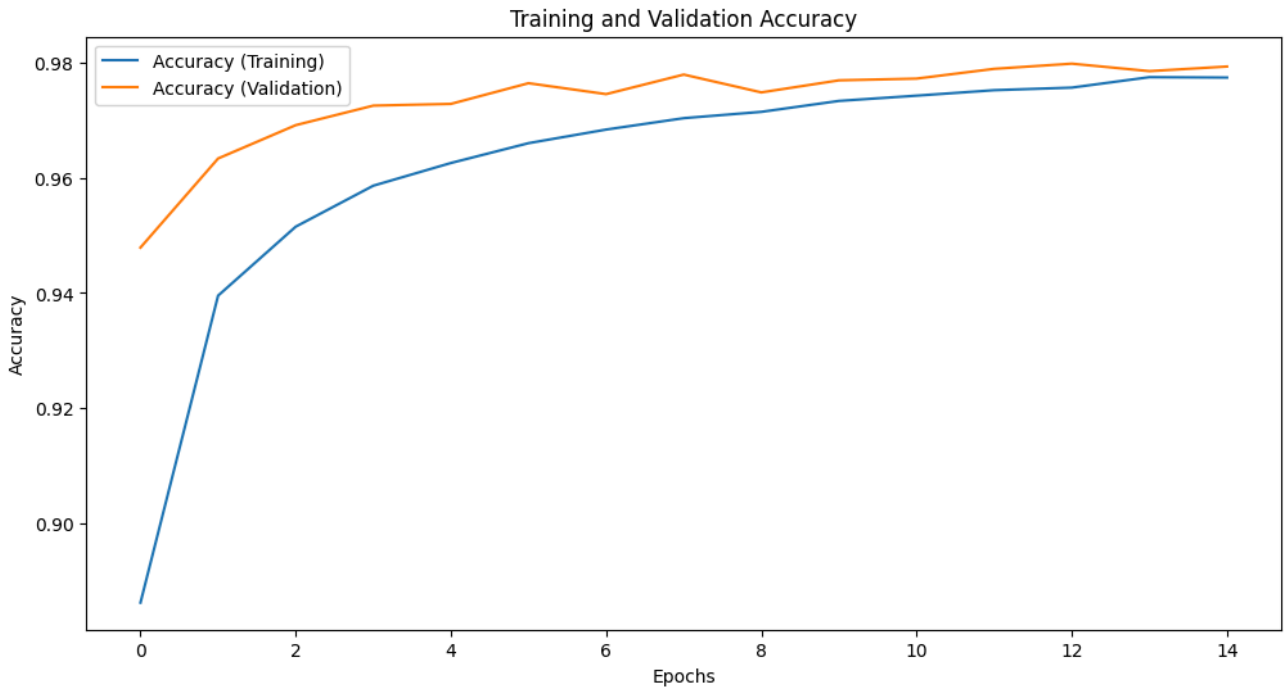

Figure 2a: Training and validation accuracy curves plotted across epochs during the training of a neural network model on the MNIST dataset with Gaussian noise.

From here, it was observed that :

Test accuracy on clean data: 0.979200005531311

Test accuracy on noisy data: 0.9793999791145325

Furthermore, signal-to-noise ratio was captured:

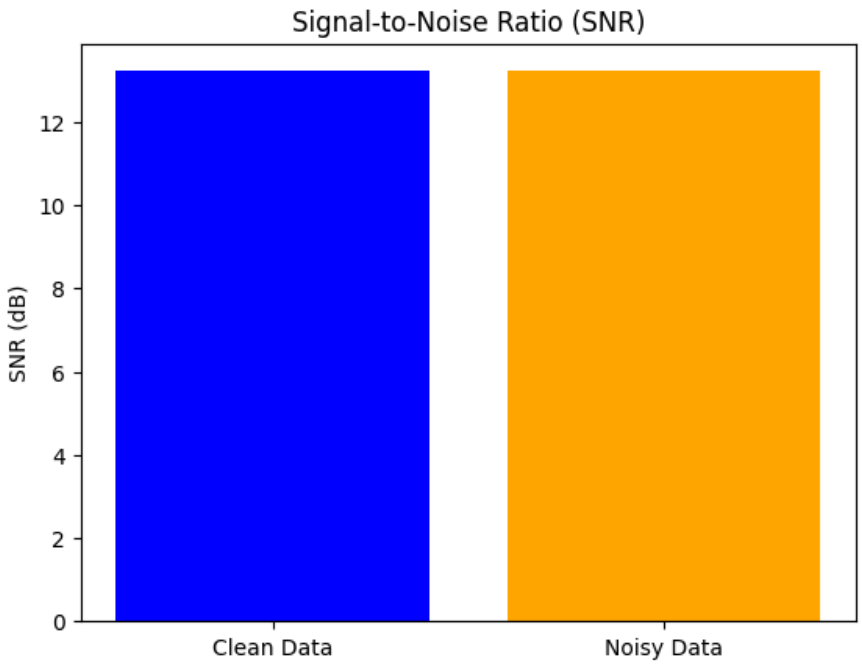

Figure 2b: Comparison of Signal-to-Noise Ratio (SNR) between Clean and Noisy Data.

SNR for clean test data: 13.219796439830542

SNR for noisy test data: 13.219796439830542

4.3 Effect of Dropout Noise on Model

Furthermore, investigations on the impact of noise on the model parameters, specifically neuronal biases, recurrent weights, and output weights, were carried out. The introduction of random noise to these parameters resulted in outcomes similar to section 4.1, characterized by significant randomization and unpredictability in the model's behavior.

The randomization of model parameters highlights the sensitivity of the model to perturbations in its internal configuration. These findings emphasize the importance of robustness and adaptability in neural ensembles, particularly in the presence of noisy environments.

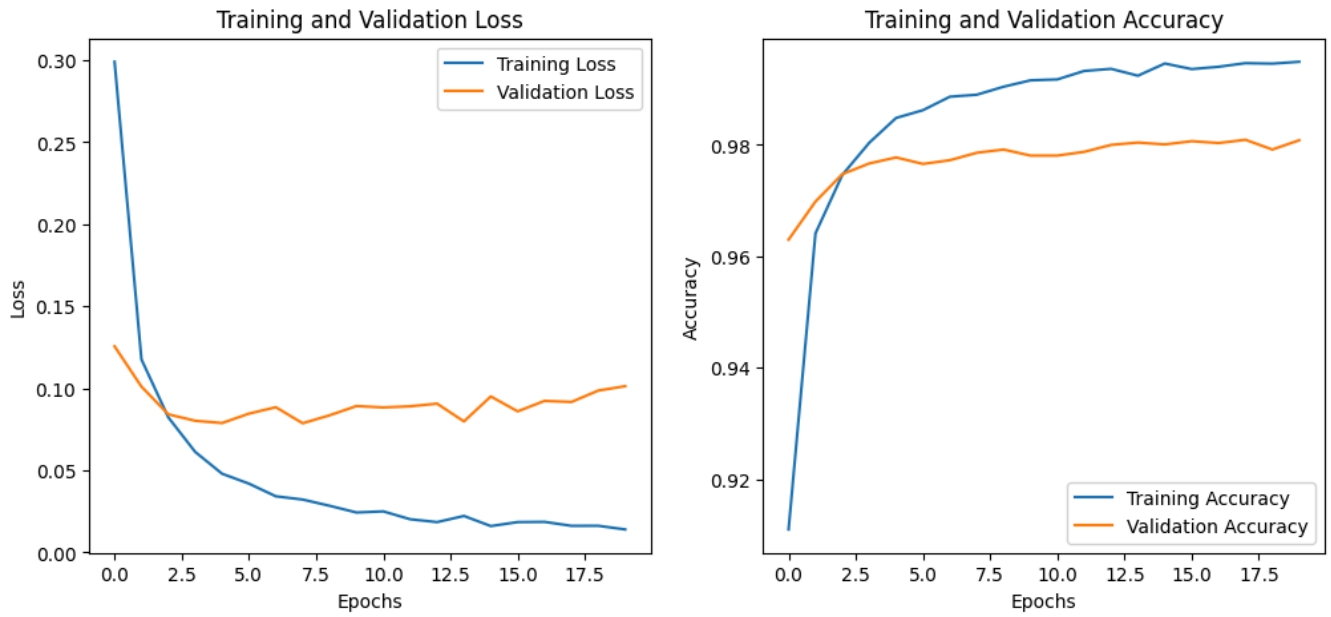

Figure 3a: Training and Validation Loss/Accuracy over Epochs

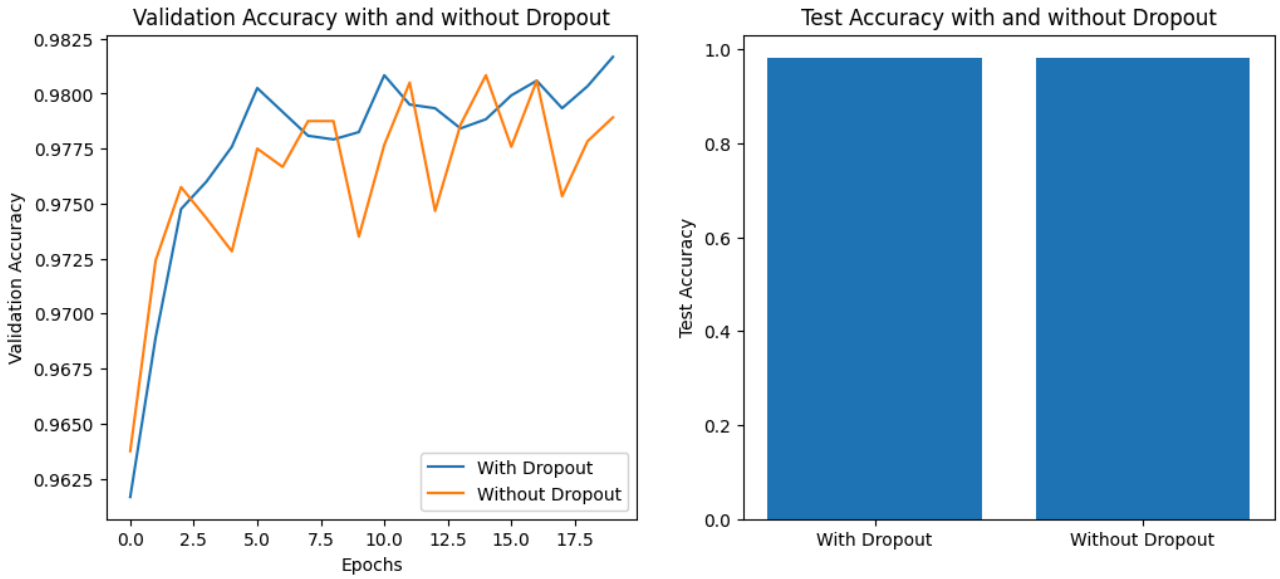

Figure 3b: Comparison of validation accuracy (left) and test accuracy (right) with and without dropout regularization over epochs. The test accuracy bar plot demonstrates the final performance of the model on a separate test dataset, highlighting the impact of dropout regularization on overall classification accuracy.

Test Accuracy with Dropout: 0.9813

Test Accuracy without Dropout: 0.9805

4.4 Effect of Adversarial Noise on Model

Adversarial noise was introduced to the model, and the following was observed.

Figure 4: Training history plot illustrating the impact of adversarial noise on model training.

Despite the introduction of adversarial noise, the model still demonstrates an increasing trend in accuracy and validation accuracy over epochs, indicating robustness to adversarial perturbations.

4.5 Gaussian Noise over multiple trials

Experimenting with gaussian noise over 10 trials using standard deviation [0.1, 1.0, 3.0] is as shown in Figure 5.

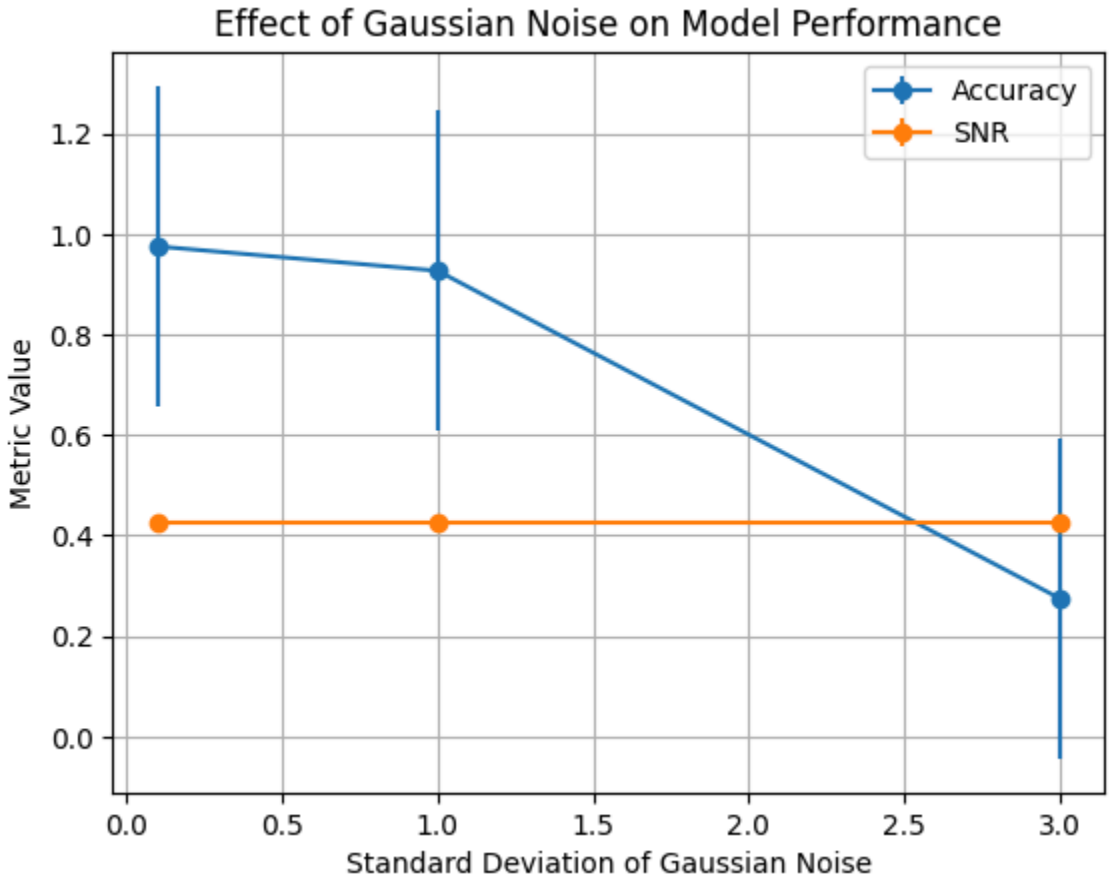

Figure 5: Effect of Gaussian Noise on Model Performance: Variation of Accuracy and Signal-to-Noise Ratio (SNR) with Standard Deviation of Gaussian Noise

4.6 Dropout Noise over multiple trials

Dropout noise over 10 trials and 10 epochs was experimented on. The plot is shown below:

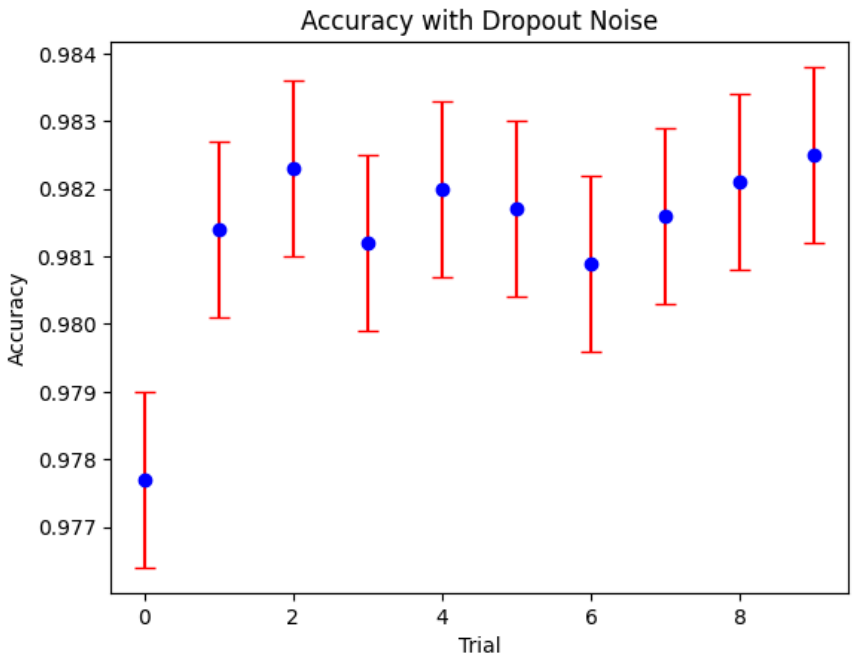

Figure 6: Variation of Accuracy Across Trials with Dropout Noise on MNIST Dataset

This produced a Mean Accuracy of 0.9813399970531463 and Standard Deviation (Accuracy) of 0.0013016935525732441.

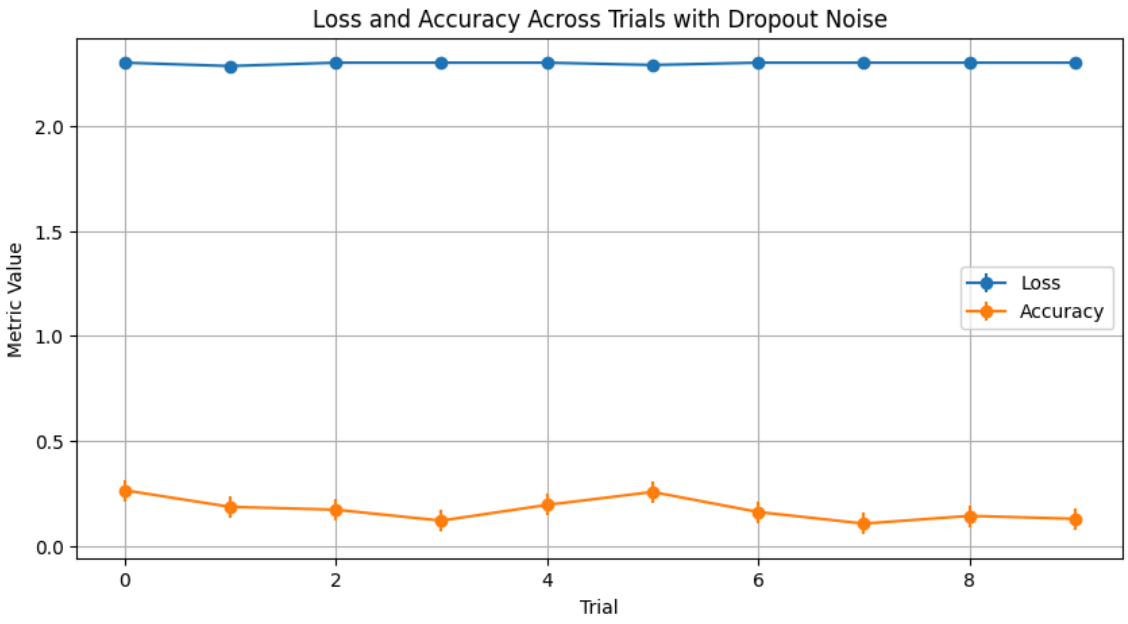

Figure 7: Variation of Loss and Accuracy Across Trials with Dropout Noise on MNIST Dataset

Further experimenting produced a Mean Loss of 2.299874043464660, Standard Deviation of Loss: 0.005550863192146892, Mean Accuracy: 0.17271999940276145 and Standard Deviation of Accuracy: 0.05142265726127164.

4.7 Adversarial Noise over multiple trials

Figure 8 below shows the effect of adversarial noise over 10 trials.

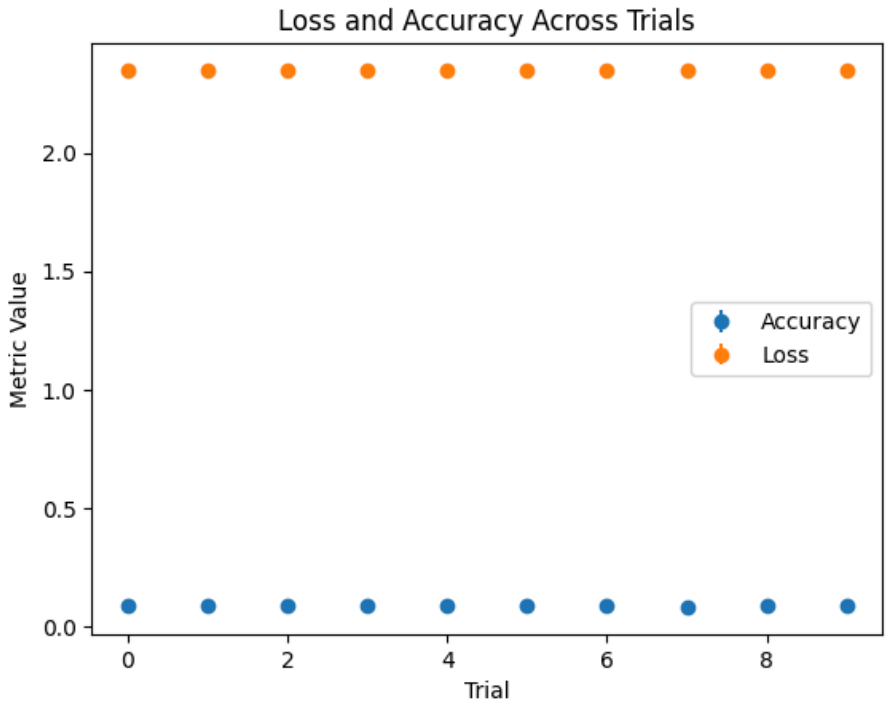

Figure 8: Variation in Loss and Accuracy Across Trials with Adversarial Noise

This experiment produced Mean Accuracy: 0.08756999894976616, Std Accuracy: 0.0014805741497860973, Mean Loss: 2.3462186813354493 & Std Loss: 0.0006524352902692283.

5. Discussions

5.1. Interpretation of Results

1. Impact of Gaussian Noise: The performance of the model was inversely related to the standard deviation of Gaussian noise added to the input data. As the standard deviation increased, both accuracy and signal-to-noise ratio (SNR) decreased. This indicates that higher levels of noise in the input data negatively affect the model's ability to make accurate predictions.
2. Impact of Dropout Noise: The accuracy of the model seemed imbalanced with the addition of dropout noise. This can be seen in the undulating nature of the accuracy plots while the loss appears stable.
3. Impact of Adversarial Noise: The adversarial noise has a negligible impact on both the loss and accuracy of the model across multiple trials. Both the loss and accuracy values remain consistent and do not show significant variation across trials, as indicated by the closely clustered orange and blue dots. The consistency of the metrics across trials suggests that the model's performance, in terms of both accuracy and loss, is robust to the presence of adversarial noise.

5.2. Comparison with Existing Literature

Previous studies, such as Zhang et al. (2019), suggested that the presence of Gaussian noise during training acts as a regularizer, improving generalization performance by reducing overfitting. From the experiments, this is accurate with small levels of gaussian noise. As the gaussian noise intensifies, the system’s accuracy is drastically reduced.

Srivastava et al. (2014) introduced dropout regularization as a technique to prevent co-adaptation of neurons, thereby enhancing generalization on image classification tasks. In the experiments, the addition of dropout noise seemed to make the system unstable, as it fluctuated.

Lastly, studies such as Goodfellow et al. (2015) highlighted the vulnerability of neural networks to adversarial attacks, which can significantly degrade generalization performance. From experiments, it seemed like the feedforward neural network is robust enough to handle the level of adversarial attack it was presented with.

5.3. Limitations and Future Directions

This experiment took a small set of data, the MNIST dataset. Future experimentation on other larger datasets would help to ascertain the observations and findings

In addition, just like for the gaussian noise, further investigation can explore higher noise levels for the dropout and adversarial noise.

5.4. Conclusion

In conclusion, the impact of noise on systems is inevitable. This study has experimented with the introduction of noise to a feedforward neural network and has studied its effects. With that in mind, neural networks mirroring biological systems should always consider noise and should be built robustly so that noise doesn’t affect its expected behaviors and outputs.

References

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

Xie, C., Wang, J., Zhang, Z., Ren, Z., & Yuille, A. (2020). Self-training with noisy student improves ImageNet classification. arXiv preprint arXiv:1911.04252.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2019). Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530.