Jestr

Journal of Engineering Science and Technology Review 16 (3) (2023) 66 - 73

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

An Experimental Study on the Deviations in Performance of FNNS and CNNS in the Realm of Grayscale Adversarial Images

Steve Mathew D. A.[1,2], Durga Shree N.[1,2] and Chiranji Lal Chowdhary[1,3,*]

[1]*Department of Software and Systems Engineering, School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India*
[2]*Data Science and Applications, Indian Institute of Technology Madras, Chennai, India*
[3]*Botho University, Botswana.*

Received 13 October 2022; Accepted 26 April 2023

Abstract

Convolutional Neural Networks, CNNs are known for their unparalleled accuracy in the classification of benign images. It is observed that neural networks are prone to having lesser accuracy in the classification of images with noise perturbation. The following study resulted in inferences to establish that CNNs are extremely vulnerable at predicting noisy images while Feed-forward Neural Networks, FNNs are least affected due to noise perturbation, maintaining their accuracy almost undisturbed. FNNs showcase better classification accuracy when tested with noise-intensive, single-channelled images that are just sheer noise to human vision. The hand-written digit images from the MNIST dataset t are classified using the architectures of FNNs with 1 and 2 hidden layers and CNNs with 3, 4, 6, and 8 convolutions, which provide the stated experimental inferences. Deviations in the performances of these architectures analyzed systematically propose that FNNs stand out to show a classification accuracy of more than 85%, irrespective of the intensity of noise and CNNs witness a trend in the reduction in speed of classification accuracy against increasing noise intensities. Correlation analysis and mathematical modelling of the accuracy trends act as roadmaps to picture that the change in the speed of classification accuracy against increasing noise intensities for CNN with 8 convolutions is half of that of the rest of the CNNs. This experimental study is a step to quantify the performance of deep learning image classification models in the context of adversarial images.

Keywords: Adversarial Examples, CNNs, FNNs, Classification Accuracy, Noise Perturbation

1. Introduction

Image classification involves training a neural network to categorize images into different classes or categories, such as identifying objects or scenes in images. Neural networks, including Feed Forward Neural Networks (FNNs) and Convolutional Neural Networks (CNNs), are widely used for image classification due to their ability to learn complex patterns from large amounts of data.

Adversarial examples are images that are created by adding noise that is specifically curated to fool machine learning models [1]. These have posed a serious threat to image classification algorithms and systems in recent times. The primary concern with regard to these adversarial examples is that they are not distinguishable (the noise added is not profound) from the benign images to the human eye. Hence, an evasion attack can be implemented against machine learning models using these adversarial images in order to fool them, resulting in misclassifications [21]. These attacks can be performed in both white-box scenarios, where the attacker has complete knowledge of the neural network's architecture and parameters, and black-box scenarios, where the attacker has limited knowledge and access to the neural network. Previous studies have trained the model using adversarial input with an attack step size of up to 16 to introduce the model to adversarial data before testing against adversarial images [2, 26]. This training teaches the model to expect changes at the pixel level, those which could be potential sources to fool the model, and act accordingly while encountering such images in the test set. The benign dataset consists of images without noise. In this study, the model is trained using the benign dataset and is tested for its learning capabilities using images severely exposed to noise distortion. This is to evaluate the model's robustness towards adversarial images encountered in real-time.

Previous works [5] have focussed on comparing and contrasting Deep Belief Networks (DBN) with CNNs under adversarial images. It was found that DBNs performed better than CNNs for adversarial examples. The work attributes the drawback of CNNs to strong inductive bias assumption, which is an attribute of the working of CNN. Alterations have been made to the CNN architecture to come up with models that improve the robustness of CNNs against adversarial examples [6]. In a unique study, CNN architecture has been altered to denoise an image before processing [7]. Previous studies have been engaged in finding the appropriate activation function for the hidden layer of an FNN. Laudani et al. also proposed a method to change a network configuration between various activation functions without affecting the network mapping capabilities [15].

Many real-life scenarios experience the threat of adversarial images which results in huge material loss and compromised decisions. Adversarial Images can fool medical diagnosis systems where something malignant could be incorrectly classified as benign or vice-versa [18]. In practice, altering pixels to fool machine learning models is common. Adding an adversarial patch that can influence the decision of

*E-mail address: prof.chowdhary@gmail.com
ISSN: 1791-2377
doi:10.25103/jestr.163.09

a model has been a new viable adversarial attack [19]. These adversarial attacks are not just reserved to image dataset classifiers. Time series classifiers, speech recognition systems, video processing systems, object detectors, etc are also vulnerable to adversarial attacks [20]. Quality, safety and security monitoring AI models are also at risk of manipulation.

Al-Shedivat et al. introduce SignSGD, a novel method for generating adversarial examples by optimizing the signs of the gradients, which is more computationally efficient compared to traditional gradient-based attacks, making it applicable to non-convex problems [29]. Also, Croce et al present a mixed integer programming-based method for evaluating the robustness of neural networks against adversarial attacks, which provides a more rigorous and scalable approach for assessing the robustness of deep learning models [30].

Neural Networks have a standard approach of learning from image datasets. Initially, neural networks assign random weights to the weight matrices that attempt to establish relationships between any two layers. Training is essentially just a process to adjust these weights into meaningful values that capture features of the training dataset. Gradient Descent method of the training process is used to find a minimum of the loss function. A global minimum cannot be guaranteed always. Loss function represents the loss incurred due to insufficient adjusting of the weights [11]. The weights at a particular instance of time can be adjusted by taking the gradient of the loss function at that point and by stepping in the negative direction of the gradient. Cross-entropy [12] is a differentiable function which accounts for providing feedback towards stepwise improvement of the model by assigning higher probability to the correct label in order to reduce loss.

While dealing with noisy images, the features captured by models do not sufficiently support the classification due to deviations from expected patterns of test inputs at the pixel level. In real-life scenarios, it is possible that the model will come across images that are distorted, noise-intensive and misleading. Such images can also be created computationally. The generation of adversarial images follows a procedure opposite to that of gradient descent. Differentiating the loss function with respect to parameters to decrease the loss on the sample is done to reach the minimum (at least local) in gradient descent. Similarly, differentiating the loss function with respect to the input data to modify the input data such that the expected loss of the model increases in the sample data generates the required sample of adversarial images. This significant relationship between model-training and model-fooling is one of the interesting aspects of exploring adversarial robustness. Fast Gradient Sign Method, FGSM, a method proposed by Goodfellow, et al, to generate adversarial inputs [9] is precisely the same as doing one gradient ascent step, with the exception that we fix the perturbation on each pixel to be a constant size - epsilon, which ensures that no pixel in the adversarial example differs from the original picture by more than epsilon.

In order to evaluate the robustness of neural networks to noise perturbation in image recognition, we add noise to normal images. Some of the techniques that are used to add noise to images are Fast Gradient Sign Method, One-step target class methods, Basic iterative method, Iterative least-likely class method etc., [2]. There are also numerous methodologies applied to defend a model from adversarial attacks. The two common approaches are, increasing robustness of machine learning models and detecting adversarial attacks before testing [14]. To increase the robustness of machine learning models, the models can be trained with adversarial examples. The other methods include defensive distillation, random resizing and padding, stochastic activation pruning, total variance minimization and quilting, thermometer encoding, adversarial logit pairing, etc. To detect adversarial attacks, the initial approaches used were principal component analysis, softmax, and reconstruction of adversarial images. Other recent techniques include feature squeezing, adversary detector networks, reverse cross-entropy, kernel density and Bayesian uncertainty estimates [14].

Images captured in real-time applications tend to be attacked by adversarial noise. Hence, image classification models that are robust to such attacks are to be identified. So, the primary question addressed here is: are CNNs, considered to be the best models for image classification, even good at classifying adversarial images? Given the ability of CNNs to classify images accurately using local spatial coherence, it is hypothesized that CNNs would be accurate in classifying adversarial images as well. But the experimental results prove otherwise.

In this study, we compared the accuracies of Feed Forward Neural Networks (FNNs) and Convolutional Neural Networks (CNNs) on adversarial test sets. FNNs and CNNs have different perspectives when processing images. While CNNs are good at handling benign datasets, FNNs show commendable stability and robustness on noisy, single-channelled images. FNNs analyze individual pixels to derive patterns for each output bin, while CNNs focus on utilizing the data of nearby pixels (local spatial coherence) to understand their associations [3][4]. The organization of neurons in the animal visual cortex, which respond to overlapping portions of the visual field, inspired the connection network of CNNs [27][28]. Both FNNs and CNNs are capable of image classification, but CNNs only consider the proximal positions of pixels, while FNNs are sensitive to the position of the object of interest in the image [31][32].

In the following sections, we perform experiments to compare the classification accuracy of different neural networks and derive interpretations from the results obtained to further understand the behaviour of the models under study for both benign and adversarial test samples. Our contributions include:

- Examining the performance trend of the neural networks of interest under adversarial attacks.
- Modelling the reduction in speed of classification accuracy against increasing noise intensities
- Hypothesizing the reason for the poor performance of CNNs against adversarial greyscale images.

2. Methods

2.1. Architecture of FNNs used in the Study

In FNNs (as shown in Figure 1), no neuron in the output layer acts as an input to a preceding layer or the same layer. Three FNNs with varying sizes and number of hidden layers are considered. Two single-hidden-layer FNNs with hidden layers of sizes 32 and 256 neurons are and an FNN with two hidden layers of 256 and 32 neurons is constructed for the study.

The dimension of each sample image in the MNIST dataset is 28x28 pixels. There are 784 input neurons which represent the 784 pixels. For instance, if we consider the model with 32 neutrons in its hidden layer, the size of the

input layer would be 1x784 and the dimensions of the weight matrix for the hidden layer would be of the size 784x32. After the data passes through the hidden layer, the weight matrix for the output layer would be of the size 32x10. When we multiply the matrix of the input layer with that of the weight matrix between the input and hidden layer, we will arrive at a matrix of size 1x32 and when the same is multiplied with the weight matrix between the hidden and the output layer, we would arrive at a matrix of size 1x10 which denotes the output of the neural network. The final 10 output values tell the probability of the test image belonging to each of the output bins. Rectified Linear Unit (ReLU), an activation function [22], is applied after each of the layers to keep the relationships between the input and output non-linear. Softmax function is applied after the last layer to assign probabilities of classification to all the bins. The equation for ReLU [22],

$$f(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} = max(0, x)$$

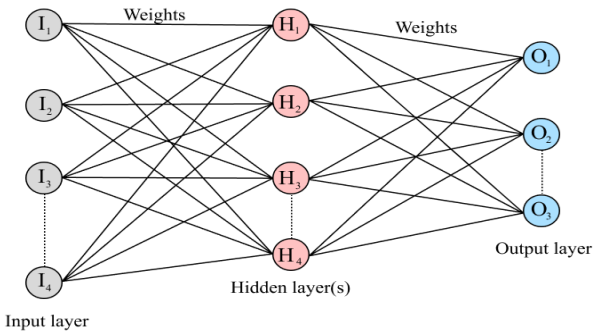

Fig. 1. A simple architecture of a Feedforward Neural Network [16].

2.2. Architecture of CNNs used in the Study

In the CNNs developed, with the architecture shown in Figure 2, a kernel of size 3 is used to traverse over the pixels in the image. A kernel is essentially a filter which is used for feature extraction. Several kernels together form a convolution that acts as the feature repository at a particular stage in the feature extraction process. Padding is the process of adding a black or white pixel around the edges of the image [23]. A padding of 1 pixel is used to make sure that the dimensions of the image remain the same even after sliding the kernels. After each of the convolutions, the activation function ReLU is used before the next convolution is performed. After every two convolutions, max-pooling is done to condense the information into smaller matrices [13].

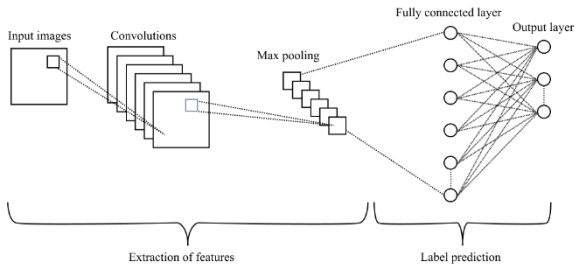

Fig. 2. A simple architecture of a Convolutional Neural Network [17].

A kernel performs the task of extracting the low-level features. Once max-pooling is applied on the convolution of low-level features, other high-level features are extracted through subsequent filters (hierarchical feature learning). Stride is the number of pixels by which the kernel is shifted each time during the traversal. When the value of the stride is very high, then the kernel 'hops' leaving many pixels in between the hops. For instance, when the kernel size is set to 3 and the stride is set to 3, then each pixel will be traversed exactly once. If the value of stride is set low, then the resolution of the filtered image will be high due to application of multiple slides of different parts of the kernel on the same section of matrix in the convolution phase.

2.3. Experimental Background

In this study, we have considered the MNIST dataset which is a curated collection of hand-written image datasets in grayscale (single-channeled) with 50000 training images and 10000 test images. An epoch is the total number of iterations required to train the machine learning model using all of the training data at once [24]. Since the accuracies of FNNs flatlined around 100 epochs and that of CNNs at 50 epochs, the models were trained and modelled for the same.

The method used to generate adversarial examples is the Fast Gradient Sign Method [9] which "linearizes the cost function to obtain an optimal max-norm constrained perturbation". FGSM focusses on adding noise whose direction is the same as the cost function's gradient in accordance with the data.

The activation function used in the study is the Rectified Linear Unit (ReLU) [10]. The output of the hidden layer and the inputs have a linear relationship ie., each element of the output of the hidden layer is the product of the weights and the elements from the input layer. Hence, a linear function of inputs has been established. This makes the hidden layers capable of only capturing the linear relationships between the input layer and the output layer. Application of ReLU, being a nonlinear function, performs activation, to capture nonlinear relationships. Loss is the penalty for a bad prediction. Technically, the loss function is generated from the difference in the expected and predicted output [25]. The computation of loss is done using the cross-entropy method. Minimization of the computed loss by administering suitable weight adjustments is implemented using the Gradient Descent Method (Optimization Algorithm).

A learning rate of 0.01 was used on FNNs throughout the training process. The sizes of the hidden layer for the FNN model with one hidden layer were set to 32 and 256. The size of the hidden layers for the FNN model with two hidden layers was set to 256 and 32. Learning rates ranging between 0.001 and 0.0000001 were used for CNNs based on the speed of reaching the minima (minimum loss). To speed up the reachability of minimum loss, learning rates were decreased gradually.

3. Results and Discussions

3.1. Performance of Models on Benign Dataset

CNNs are widely used for image classification and known for their accuracy in effectively classifying images with multiple channels. But FNNs are not as popular as the former in the realm of image classification. This is because CNNs take into account the relative position of pixels in an image when performing feature extraction. But in FNNs, the relative positions are not taken into account and the images are just seen as a set of pixels.

FNNs show comparable accuracy with CNNs in classification due to the simplicity of the dataset. In the

dataset of interest (MNIST), each sample image is gray-scaled and is composed of 784 pixels (28 X 28). The sample images of such minimal resolution are not comparable with those found in real-life. Owing to this low resolution and single-channeled input, FNNs were capable of classifying these images to a good degree of accuracy and so FNNs were considered in this study to be compared with the CNNs.

All models in Table 1 show acceptable accuracy in classification of benign images. Even when the architectures of the models were altered by increasing the number of layers or convolutions, classification accuracy shows no significant change. Owing to the smaller number of pixels in each training image and single-channelled input, the models converge quickly.

Table 1. Classification accuracies of the models for benign test images.

Models	FNN (1 - hidden layer)	FNN (2 - hidden layers)	CNN (3 - convolutions)	CNN (4 - convolutions)	CNN (6 - convolutions)	CNN (8 - convolutions)
Accuracies	96.20%	97.70%	99.47%	99.52%	99.47%	99.45%

3.2. Influence of the Size of Hidden Layer on Classification Accuracy

In an FNN with 1 hidden layer, when the size of the hidden layer is increased from 32 to 256 neurons, a massive improvement is observed in the accuracy at which the model classifies adversarial images (Table 2). The model with 32 neurons in the hidden layer shows a decline in the accuracy as the attack step size increases. Its accuracy for minimal noise perturbation (epsilon = 0.1) is also not satisfactory. But the one with 256 neurons in the hidden layer has a classification accuracy with a standard deviation close to 1 and arithmetic mean being 88%. But this increase from 32 neurons to 256 did not have a big impact on the benign test examples but rather increased the accuracy of classifying adversarial images by a large margin.

Table 2. Classification accuracies of FNN with 1 hidden layer of sizes 32 and 256 against different intensities of noise.

Attack Step Size (Epsilon)	Size of hidden layer = 32	Size of hidden layer = 256
0.1	39.11%	88.69%
0.2	24.54%	88.23%
0.3	25.21%	87.03%
0.5	23.44%	88.97%
0.75	19.73%	88.51%
1	18.80%	88.61%
1.5	17.57%	88.87%
2	19.99%	88.08%
5	16.80%	88.97%
10	24.15%	86.45%
16	21.23%	86.20%

As the size of the hidden layer increases, feature extraction is effective. This can be reasoned out by the presence of a bigger weight matrix between the preceding layer and the current hidden layer of interest. These effectively extracted features make the model robust to noisy input.

3.3. Performance of Models on Adversarial Dataset

A significant amount of deviation in the performance of CNNs on adversarial data from the previous test result on benign dataset is observed. FNNs show a high and steady accuracy to all the tested range of attack step sizes. The performance of FNNs with one hidden layer of 256 neurons on adversarial images is commendable. Adding another layer of size 32 has no significant effect on improvement in accuracy. A trend of decrease in accuracy is observed in every other model except FNNs, which show no trend of variation. The accuracy ranges around 88% for FNN with one hidden layer (of size 256 neurons) and 90% for FNN with two hidden layers (of sizes 256 and 32 neurons) for the two models with just 2% of deviation within the tested attack step sizes.

The probability of guessing a number right without any training is 0.1 (10%) due to the presence of 10 possible classes each denoting a number. CNNs with 3, 4 and 6 convolutions show no better performance than guessing, with epsilon increasing beyond a threshold. This observation can be attributed to the "perplexed" state the model is in due to immense noise accumulation. Kernels act as filters to extract various features of an image at various levels [8]. Huge amounts of distortion in the test image do not give information about features the kernel is expecting to extract. Since the model is bound to classify the image into one of the ten bins, it classifies it based on what it considers to be the nearest fit. The model does not have any strong cause as to why it is classifying an image into a particular bin and hence the guessing process.

3.4. Correlation Between the Accuracies shown by Various Models for Tested Epsilons (r^2)

The correlation between the range of accuracies of FNN (1 - hidden layer) and FNN (2 - hidden layers) cannot be commented on or compared with other models because the accuracies of the models do not follow a trend of inclination or declination. The accuracies remain approximately steady for the entire range of noise addition. The correlation among all possible combinations of the analyzed CNNs is high and it shows that the trend of decline of their accuracies across the range of attack step sizes are similar. Among them, the correlation between the CNNs with 3, 4 and 6 convolutions are very high and their trends of decline are identical to each other.

From table 3, it can be inferred that the accuracy is the maximum for CNN with 8 convolutions when the attack step size is very low (in our case, 0.1) even when compared to other FNN models. Other CNN models don't correlate with CNN with 8 hidden layers as much as they do among themselves. This slight variation in correlation is due to the slightly better performance of CNN (8 convolutions) when compared with other CNNs when the attack step size is increasing beyond 0.5 (Table 4).

Table 3. Classification accuracies of networks constructed with different intensities of noise perturbation.

	Classification Accuracies					
Attack Step Size (Epsilon)	**FNN(1 hidden layer)**	**FNN(2 hidden layers)**	**CNN(3 convolutions)**	**CNN(4 convolutions)**	**CNN(6 convolutions)**	**CNN(8 convolutions)**
0.1	88.69%	94.02%	94.50%	90.15%	94.10%	96.47%
0.2	88.23%	91.05%	70%	65.79%	75.50%	83.76%
0.3	87.03%	87.66%	37.50%	38.66%	51.00%	79.11%
0.5	88.97%	91.35%	14.50%	20.19%	23.02%	65.15%
0.75	88.51%	88.58%	19.20%	14.65%	14.93%	38.14%
1	88.61%	89.47%	16.39%	15.67%	14.29%	40.38%
1.5	88.87%	89.37%	15.24%	20.44%	11.15%	38.08%
2	88.08%	90.66%	19.56%	16.79%	14.56%	30.04%
5	88.97%	88.79%	12.61%	11.58%	19.29%	32.10%
10	86.45%	89.30%	17.40%	13.83%	17.80%	31.21%
16	86.20%	91.74%	19.17%	10.77%	18.48%	16.73%

Table 4. Correlations between the trend of decline of classification accuracies of networks for adversarial images.

Correlation of Accuracies	**FNN (1 - hidden layer)**	**FNN (2 -hidden layers)**	**CNN (3 - convolutions)**	**CNN (4 - convolutions)**	**CNN (6 - convolutions)**	**CNN (8 - convolutions)**
FNN (1 hidden layer)	1	0.09407	0.09382	0.18296	0.07111	0.27618
FNN (2 hidden layers)		1	0.61470	0.58903	0.53898	0.35413
CNN (3 convolutions)			1	0.98765	0.97756	0.82924
CNN (4 convolutions)				1	0.97819	0.88976
CNN (6 convolutions)					1	0.89334
CNN (8 convolutions)						1

3.5. Visual Representation and Interpretation with Respect to Accuracies

The benign images when added with the carefully curated noise resulted in the adversarial images in Fig. 3, 4 and 5. The attack step size (epsilon) denotes the intensity of noise added to the image.

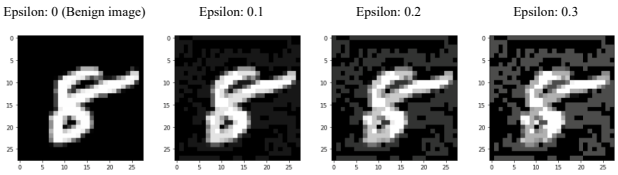

Fig. 3. Adversarial examples with lower noise intensities.

In Fig. 3, it can be observed that the number in the image is evident to the naked eye. But the addition of noise is also not unnoticeable. The minimal noise added is also noticeable in this dataset because of the lower resolution of the images.

Fig. 4. Adversarial examples with high noise intensities.

In Fig. 4, the concentration of noise is so high such that the number in the image is not identifiable to the human eye. When the concentration of the noise reaches an attack step size of 1, the image just becomes sheer noise and no observation can be drawn from the image for a human.

Fig. 5. Adversarial examples with extremely high noise intensities.

The levels of attack step-size in Fig. 5 do not have such huge influence in distortion in real-life multi-channeled images. Since the image resolution of this dataset is quite minimal (28x28), these attack step sizes have a drastic effect on the image.

The attack step size of 1 serves as a breaking point for this dataset such that any amount of noise above this limit is resulting in an image that looks the same. This could act as a potential point that challenges the robustness of the CNN models towards adversarial noise. Just as it is evident to the human eye that no peculiar pattern can be observed from any of the images after this point, the models also show very identical 'guessing' behavior (10%-15% correctness in classification).

3.6. Mathematical Modelling of the Trend in Accuracies shown by CNNs

In this section, we attempt to fit a mathematical equation to the trend followed by the considered FNN and CNN models.

3.6.1. For an Attack Step Size considered up to 16

The squared correlation from table 5 doesn't show many positively correlated model accuracies to generate nearest fits. The only two noticeable fits are:

- CNN (8 - convolutions) which can be fit into the logarithmic equation, $f(x) = 0.524 - 0.149 \ln(x)$
- CNN (8 - convolutions) which can also be fit into the power series equation, $g(x) = 0.463x^{-0.313}$

Except for the above two equations, the others are not likely reliable owing to their small values of correlation. Thus, we attempt to further the study by modelling the data only up to an attack step size of 2 so that the classification accuracies of higher noise intensities that result only in the models 'guessing' the output are omitted.

Table 5. Squared correlation values of the model fits for the decline of classification accuracies of networks for attack step sizes up to 16.

Attack step-size<=16	Linear	Exponential	Logarithmic	Polynomial	Power Series
CNN(3 convolutions)	0.112	0.066	0.509	0.264	0.513
CNN(4 convolutions)	0.189	0.209	0.613	0.325	0.744
CNN(6 convolutions)	0.123	0.07	0.541	0.267	0.49
CNN(8 convolutions)	0.409	0.482	0.834	0.531	0.879

3.6.2. For an Attack Step Size considered up to 2

As mentioned in Section 3.4, we cannot expect a deviation trend from FNNs. We can approximately fit them into the following lines:
FNN (1 - hidden layer) is following a linear equation,

$$p(x) = 88.08$$

FNN (2 - hidden layer) is following a linear equation,

$$q(x) = 90.18$$

The squared correlation (r^2) is tabulated to understand the closeness of fits generated with the actual trend curves of the models (Table 6).

An attempt to fit the decline of classification accuracies of the networks against attack step sizes using linear, exponential, polynomial, logarithmic and power series equations is made.

Table 6. Equations of the model fit for the decline of classification accuracies of networks for attack step sizes up to 2.

Attack step-size<=2	Linear	Exponential	Logarithmic	Polynomial	Power Series
CNN (3 convolutions)	$-0.294x + 0.592$	$0.757e^{-1.02x}$	$0.197 - 0.258 \ln x$	$0.896 - 1.34x + 0.512x^2$	$0.187x^{-0.623}$
CNN (4 convolutions)	$-0.283x + 0.577$	$0.725e^{-0.999x}$	$0.2 - 0.245 \ln x$	$0.854 - 1.23x + 0.466x^2$	$0.19x^{-0.612}$
CNN (6 convolutions)	$-0.362x + 0.66$	$0.858e^{-1.29x}$	$0.189 - 0.294 \ln x$	$0.973 - 1.44x + 0.528x^2$	$0.166x^{-0.779}$
CNN (8 convolutions)	$-0.332x + 0.853$	$0.932e^{-0.662x}$	$0.439 - 0.239 \ln x$	$1.03 - 0.936x + 0.296x^2$	$0.418x^{-0.414}$

Table 7. Squared correlation values of the model fits for the decline of classification accuracies of networks for attack step sizes upto 2.

Attack step-size <=2	Linear	Exponential	Logarithmic	Polynomial	Power Series
CNN(3 convolutions)	0.427	0.607	0.774	0.794	0.766
CNN(4 convolutions)	0.455	0.638	0.802	0.805	0.813
CNN(6 convolutions)	0.566	0.803	0.881	0.907	0.908
CNN(8 convolutions)	0.773	0.879	0.941	0.947	0.919

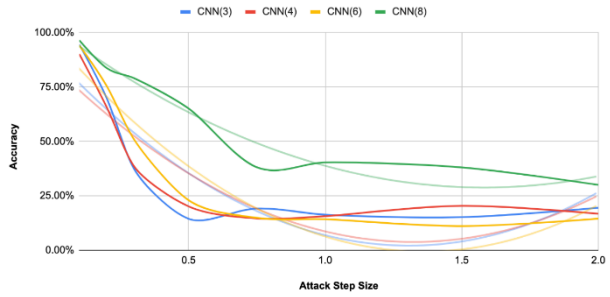

Fig. 6. Plot of decline in classification accuracies against the attack step sizes for the networks along with suitable polynomial fits

The r^2 values for logarithmic, polynomial and power series fit are comparable and acceptable due to the presence of positive correlation between the model-fit generated and available data curve (Table 7). Since the slope of logarithmic and power series models are asymptotic, the polynomial model is used for further study. It is important to note that the polynomial fits proposed are only applicable up to the point where the polynomial attains its minimum. This can be explained as the polynomial function increases after reaching its minimum but the accuracy of classification keeps declining. We can see that all the polynomial fits attain their minimum between an attack step size of 1 and 1.5. This goes

in accordance to our observation in section 3.5 where any amount of noise beyond an attack step size of 1 did not make a big difference to the accuracy of the models and that the point served as a breaking point of the classification accuracy of the models.

Table 8 shows the correlation values of the polynomial fits for the decline of accuracies of CNN models with attack step size being less than or equal to 2 (beyond which the accuracy trend is staggered). This gives us a clearer picture of the extent of correlation between the experimental results and computationally computed polynomial fits.

Table 8. r (correlation) for polynomial fits of the networks for attack step sizes up to 2.

Attack step-size <=2	Correlation value
CNN(3 convolutions)	0.891
CNN(4 convolutions)	0.897
CNN(6 convolutions)	0.952
CNN(8 convolutions)	0.973

The equations 1, 2, 3 and 4 represent the polynomial fits for CNNs with 3, 4, 6 and 8 convolutions respectively.

$$a(x) = 0.51x^2 - 1.34x + 0.896 \quad (1)$$

$$b(x) = 0.466x^2 - 1.23x + 0.854 \quad (2)$$

$$c(x) = 0.528x^2 - 1.44x + 0.973 \quad (3)$$

$$d(x) = 0.296x^2 - 0.936x + 1.03 \quad (4)$$

The obtained polynomial equations are differentiated to find the rate at which the classification accuracies of the models are declining.

$$a'(x) = 1.02x - 1.34 \quad (5)$$

$$b'(x) = 0.932x - 1.23 \quad (6)$$

$$c'(x) = 1.056x - 1.44 \quad (7)$$

$$d'(x) = 0.592x - 0.936 \quad (8)$$

The above functions are the slopes of the polynomial fits with x denoting the attack step size. They show the rate at which the accuracy declines. Differentiating it one more time would give us a value which denotes the slope of the rate of declination or the deceleration of accuracy when the attack step size increases.

In Figure 7, lines that represent the deceleration of accuracies of models under adversarial noise are plotted and it is computed that the accuracy of CNN with 8 convolutions decelerates at a slower pace.

Table 9. Table denoting the rate of declination and the deceleration of accuracy for the CNNs developed under adversarial noise.

Model	Rate of declination of accuracy	Slope of rate of declination (deceleration of accuracy)
CNN (3 convolutions)	$1.02x - 1.34$	1.02
CNN (4 convolutions)	$0.932x - 1.23$	0.932
CNN (6 convolutions)	$1.056x - 1.44$	1.056
CNN (8 convolutions)	$0.592x - 0.93$	0.592

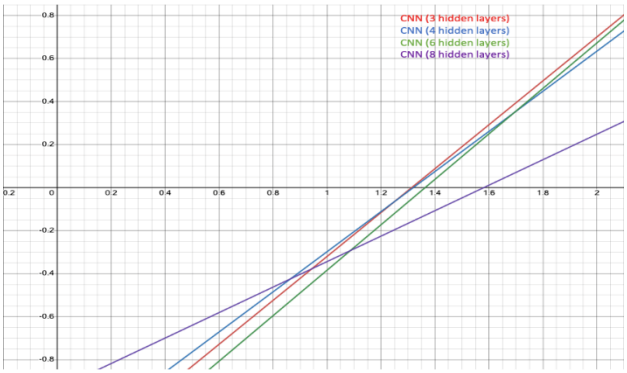

Fig. 7. Slopes of the polynomial fits that represent the decline of accuracies of the models for attack step sizes lesser than or equal to 2.

It is observed that the slope of the rate of declination of CNN with 8 convolutions is approximately half of that of the slope of rate of declination of CNNs with 3,4,6 convolutions. CNN with 8 convolutions has better accuracies even at higher attack step sizes when compared to other CNNs. This is corroborated by the value of the slope of rate of declination which is twice as much in the other CNNs as it is in the CNN with 8 convolutions.

We can observe that the line denoting the rate of declination of accuracy of CNN with 8 convolutions intersects with the lines denoting the rate of declination of accuracy of CNN with 3 and 4 convolutions between attack step size 0.8 and 1.0 and intersects the line denoting the rate of declination of accuracy of CNN with 6 convolutions between attack step size 1.0 and 1.2. At these points, the rate of declination of the corresponding models for adversarial data can be interpreted to be equal.

As discussed in Section 3.5 about attack step size of 1.0 acting as a breaking point, here it is also noticeable that it unifies the rate of declination of accuracy of all CNN models around itself.

3.7. Better Performance of FNNs than CNNs on Grayscale Adversarial Images

Recalling the architecture of FNNs, the relative positions between pixels is discarded while interpreting the relationships between the input and output vectors. Every pixel from input is individually analyzed and its significance is mapped using the adjusted weight matrix.

Preserving the local spatial coherence is a prominent principle of CNNs. Various filters extract features and this collection is condensed (max-pooled) a few times to extract higher-level features each time. It is hypothesized that distorted features are presented to kernels while applying filters and when the convolutions are max-pooled, the noisy information gets accumulated and is carried through all the hierarchical layers. The distorted information of the image available at the highest feature extraction level is due to cumulative addition of inappropriate features right from the lower-level leading to relatively poor performance in classification accuracy.

4. Conclusion

In this paper, we have compared and contrasted the performance of CNNs and FNNs under adversarial noise for grayscale images and concluded that FNNs are much more

robust to noise perturbation than CNNs. The reason for the same is also hypothesized by carefully considering the architecture of the models. The correlations between the trends of decline of accuracies among the models when adversarial images are presented are also considered to comment on the similarities in the behaviors of models. The trends of decline in accuracy were captured using different mathematical models and the most suitable among them were employed. Using the models that approximate the real trend of accuracies, we attempted to understand the rate at which the accuracy drops for each of the models and qualitatively commented on which model exhibits relatively better robustness towards adversarial noise. This helped us to gain a better understanding as to for which concentration of noise the models behave similarly and the breaking point of the CNNs under adversarial noise and hypothesize the reason for the changing classification rates.

5. Future Scope

An interesting problem statement to explore would be to understand how FNNs are able to classify the images of the MNIST dataset with an appreciable level of accuracy when the attack step size increases beyond 1 and no substantial pattern is decipherable to the human eye. The exploration can be extended to other grayscale image datasets. Multi-channeled image datasets should also be considered to quantify the robustness of image classification models, which are beyond the scope of this study. An advanced study with other forms of adversarial attacks could yield more insights.

References

1. Szegedy C., Zaremba W., Sutskever I., Bruna J., Erhan D., Goodfellow, I. Fergus, R., “Intriguing properties of neural networks”. *arXiv preprint arXiv:1312.6199*, 2013.
2. Kurakin A., Goodfellow I., Bengio S., “Adversarial machine learning at scale”. *arXiv preprint arXiv:1611.01236*, 2016
3. Svozil D., Kvasnicka V., Pospichal J. “Introduction to multi-layer feed-forward neural networks”. *Chemometrics and intelligent laboratory systems*, 39 (1), 1997, pp. 43-62.
4. Fukushima K., "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.", *Biological cybernetics*, 36 (4), 1980 pp.193-202.
5. Yang T., Silver L. D., "The Disadvantage of CNN versus DBN Image Classification Under Adversarial Conditions." In: *Canadian Conference on AI*, Canada, 2021, pp. 1-6.
6. Jin J., Dundar A, Culurciello E., "Robust convolutional neural networks under adversarial noise." *arXiv preprint arXiv:1511.06306*, 2015.
7. Divakar N. R. Venkatesh B., "Image denoising via CNNs: An adversarial approach." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Honolulu, 2017, pp. 80-87.
8. Zayed R. M., "Effect of kernel size on Wiener and Gaussian image filtering." *TELKOMNIKA (Telecommunication Computing Electronics and Control),* 17 (3), 2019, pp.1455-1460.
9. Goodfellow I. J., Shlens J., Szegedy C., "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572*, 2014.
10. Nair V., Hinton G., "Rectified linear units improve restricted boltzmann machines." In: *Proceedings of the 27th international conference on machine learning (ICML-10)*, Haifa. 2010, pp. 807-814.
11. Ruder S., "An overview of gradient descent optimization algorithms." *arXiv preprint arXiv:1609.04747,* 2016.
12. De Boer D. P. T, Mannor S. and Rubinstein Y. R., "A tutorial on the cross-entropy method." *Annals of operations research*, 134 (1), 2005, pp.19-67.
13. Dhruv P., Subham N., "Image classification using convolutional neural network (CNN) and recurrent neural network (RNN): a review." *Machine learning and information processing*, 2020, pp.367-381.
14. Al-Sakkaf S., Kassas M., Khalid M. and Abido, M. A. "Adversarial examples in modern machine learning: A review." *arXiv preprint arXiv:1911.05268*, 2019.
15. Laudani A., Lozito G. M., Fulginei R. F., Salvini A., "On training efficiency and computational costs of a feed forward neural network: a review." *Computational intelligence and neuroscience*, 2015, pp.83-83.
16. Chaudhary A., Manisha S.. "Multilayer Neural Network Design for the Calculation of Risk Factor Associated with COVID-19", *Augmented Human Research* 6, 2021, pp.1-17.
17. Phung V. H., Eun J. R., "A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets." *Applied Sciences,* 9(21), 2019, pp. 4500.
18. Finlayson S. G., Bowers J. D., Joichi I., Zittrain J., Beam A. L., Kohane S. L., "Adversarial attacks on medical machine learning." *Science,* 363(6433), 2019, pp. 1287-1289.
19. Thys S., Van Ranst W, Goedemé T., "Fooling automated surveillance cameras: adversarial patches to attack person detection." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, Long Neacg, 2019, pp. 49-55.
20. Ren H., Teng ., Hongyang Y., "Adversarial examples: attacks and defenses in the physical world." *International Journal of Machine Learning and Cybernetics* 12(11), 2021, pp. 3325-3336.
21. Battista B., Corona I., Maiorca D., Nelson B., Šrndić N., Laskov P., Giacinto G., Roli F., "Evasion attacks against machine learning at test time." In: *Machine Learning and Knowledge Discovery in Databases: European Conference,* Prague, Czech Republic, 2013, 3 (13), pp. 387-402.
22. Hahnloser R. HR, Rahul S., Misha A. M., Rodney J. D., Seung S., "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit." *nature* 405 (6789), 2000 pp.947-951.
23. Dwarampudi M., Reddy N. V. "Effects of padding on LSTMs and CNNs." *arXiv preprint arXiv:1903.07288*, 2019.
24. Carney J. G., Pádraig C., “The epoch interpretation of learning” *Trinity College Dublin, Department of Computer Science,* 1998.
25. Ketkar N. "Stochastic gradient descent." *Deep learning with Python: A hands-on introduction*, 2017, pp.113-132.
26. Ren K., Tianhang Z., Zhan Q., Xue L., "Adversarial attacks and defenses in deep learning." *Engineering,* 6 (3), 2020, pp.346-360.
27. Hubel D. H., Torsten N. W., "Receptive fields and functional architecture of monkey striate cortex." *The Journal of physiology,* 195(1), 1968, pp.215-243.
28. Fukushima K., Sei M., "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition." *Competition and cooperation in neural nets*, Springer, Berlin, Heidelberg, 1982, pp. 267-285.
29. Bernstein J., Yu-Xiang W., Kamyar A., Animashree A. "signSGD: Compressed optimisation for non-convex problems." In: *International Conference on Machine Learning,* Stockholm, 2018, pp. 560-569
30. Tjeng V., Kai X., Russ T.. "Evaluating robustness of neural networks with mixed integer programming." *arXiv preprint arXiv:1711.07356*, 2017.
31. Chowdhary C. L., Goyal A., Vasnani B. K. “Experimental assessment of beam search algorithm for improvement in image caption generation”. *Journal of Applied Science and Engineering*, 22(4), 2019, pp. 691-698.
32. Chowdhary C. L., Acharjya, D. P. “Segmentation and feature extraction in medical imaging: a systematic review”. *Procedia Computer Science*, 167, 2020, pp. 26-36.