Statistical Methods in AI

UNDERSTANDING DEEP LEARNING REQUIRES RETHINKING GENERALIZATION

Do Deep Neural Networks actually learn?

TEAM SAMURAI

Sarthak Bansal
2021101134

Krishna Singh
2021112005

Mentor TA-
Tathagato Roy

Motivation

What is it that distinguishes neural networks that generalize well from those that don't?

- A satisfying answer to this question would not only help to make neural networks more interpretable, but it might also lead to more principled and reliable model architecture design

- Traditional wisdom leads us to believe that over parameterization causes over-fitting of models but on the other hand, these large parameterized neural networks show marginal difference in train and test performance.

- Regularization techniques used during training are largely attributed to this performance. However, through systematic and extensive experimentation we show that these models fit noisy data (in both labels and images) to the extent of 100% train accuracy.

EXPRESSIVITY OF NEURAL NETWORKS

“A very simple two layer ReLU neural network can express any arbitrary labelling of the training data.”

Theorem 1*. *There exists a two-layer neural network with ReLU activations and 2n+d weights that can represent any function on a sample of size n in d dimensions.*

To empirically verify the above theorem , we perform *randomization tests** on a 3-layer MLP with ReLU activations :*

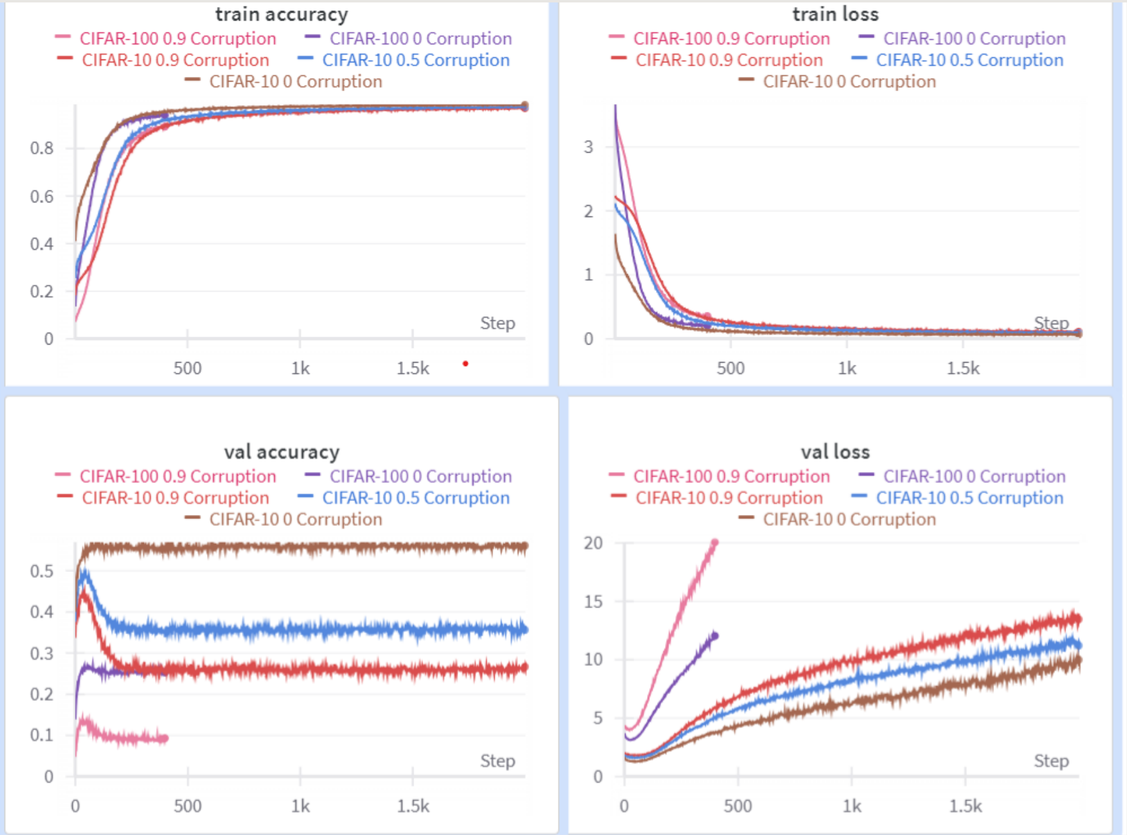
- *We train standard architectures on CIFAR-10 and CIFAR-100 datasets by injecting random labels with varying corruption levels*
- *In seperate experiments , we also introduce gaussian noise in the image*
- *We shuffle the pixels of the image , both by a pre-decided permutation as well as a random permutation for each image*

---

*Proof given in appendix : Intuition lies in the interleaving property and has no restriction on width of the network. It computes the nodes required in each layer and makes predictions based on the interval in which the value is encountered

**Randomization tests from non-parametric statistics (Edgington & Onghena, 2007)

Corrupted Labels



As one can see , a simple MLP can express any labelling of data as the train loss approaches 0 for all the cases whether noisy or not with training accuracy approaching 100%.

Corrupted Images

Even after replacing true images by Random Pixels , Shuffled Pixels (Image coherent noise) and Gaussian Noise (Image independent noise) , a 2-layer MLP fits the data and approaches 0 training loss or 100% training accuracy

---

Learning rate hyperparameter tuning done in the mid eval.

Interpretation of results

Randomizing labels and Degrading Images is solely a data transformation, leaving all other properties of the learning problem unchanged. However, the above experiments show that by doing such augmentations alone we can force the generalization error of a model to jump up considerably without changing the model, its size, hyperparameters, or the optimizer.

Universal approximation theorem* states that

$$For\ every\ \hat{f} \in C(K)\ with\ K \subset R^d\ and\ every\ \varepsilon > 0\ ,there\ exists\ a\ neural\ network\ \hat{f}_{H,\varepsilon}\ such\ that\ |\hat{f}_H - \hat{f}|_\infty \leq \varepsilon.$$

It states that even a simple neural network with depth-2 and a suitable activation function can approximate any modeling function given an optimal learning rate. The previous experiments verify the theorem.

This also establishes the fact that effective capacity of neural networks is sufficient for **memorizing** the entire data set !!

---

*G¨uhring et Raslan, Expressivity of Deep Neural Networks(2020)

Role of Explicit Regularization

“Explicit regularization techniques (dropout, weight decay, etc.) are largely credited with the success of generalization in over-parameterized neural networks.”

We saw above that model architecture alone doesn’t improve generalization performance on noisy datasets , then the question remains : **What does ?**

Explicit regularization techniques are those specifically and solely designed to constrain the effective capacity of a given model in order to reduce overfitting. Furthermore, explicit regularizers are not a structural or essential part of the network architecture, the data or the learning algorithm and can typically be added or removed easily.

In order to explore this idea, we perform randomization tests on models trained with explicit regularization methods: **Dropout** and **Weight Decay**.

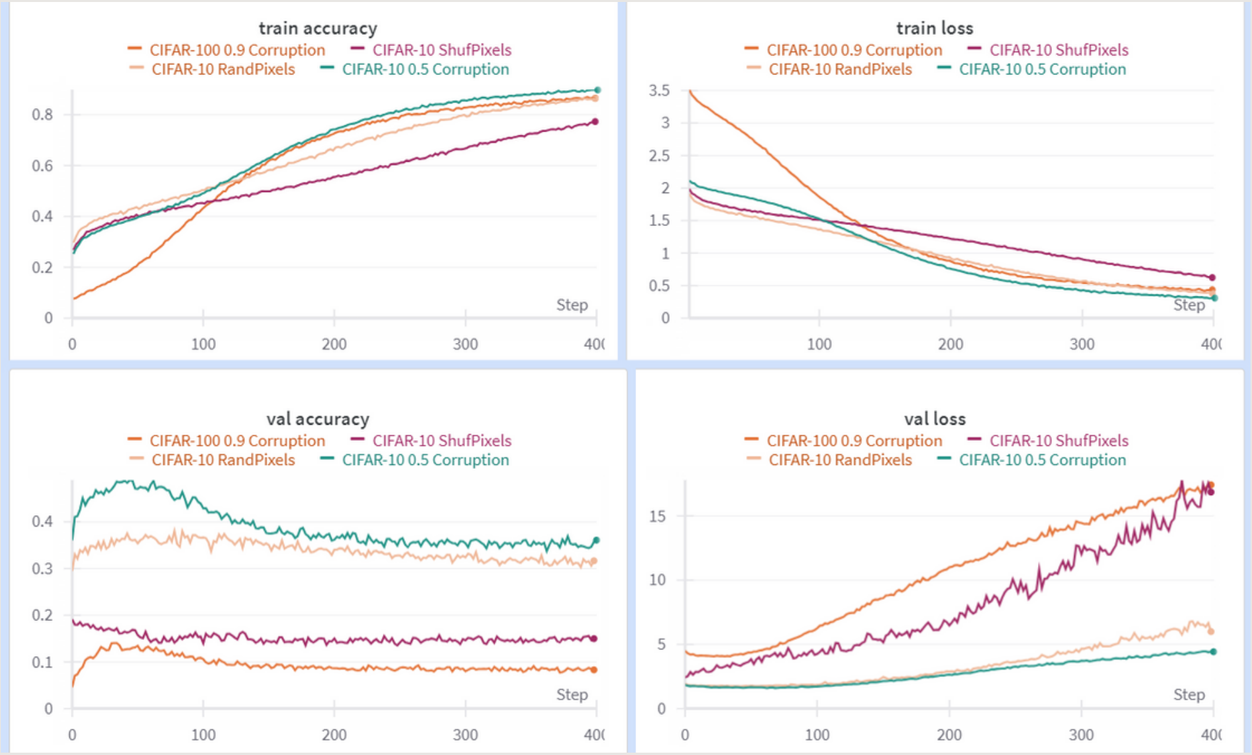
Dropout

Dropout was first described in *(1)** with the key idea being to randomly "*drop*" units from the neural network during training.

It hypothesized a way of approximately combining exponentially many different neural networks providing a smaller hypothesis class to search on.

Dropout also simulates a sparse activation from a given layer, which interestingly, in turn, encourages the network to actually learn a sparse representation as a side-effect.

Experimentation



When experimented with different types of noise on CIFAR-100 and CIFAR-10, we observe that the model still approaches 0 training loss / 100% training accuracy .

*Nitish Srivastava, Alex Krizhevsky et al., Dropout: A Simple Way to Prevent Neural Networks from Overfitting (2014)

Weight Decay

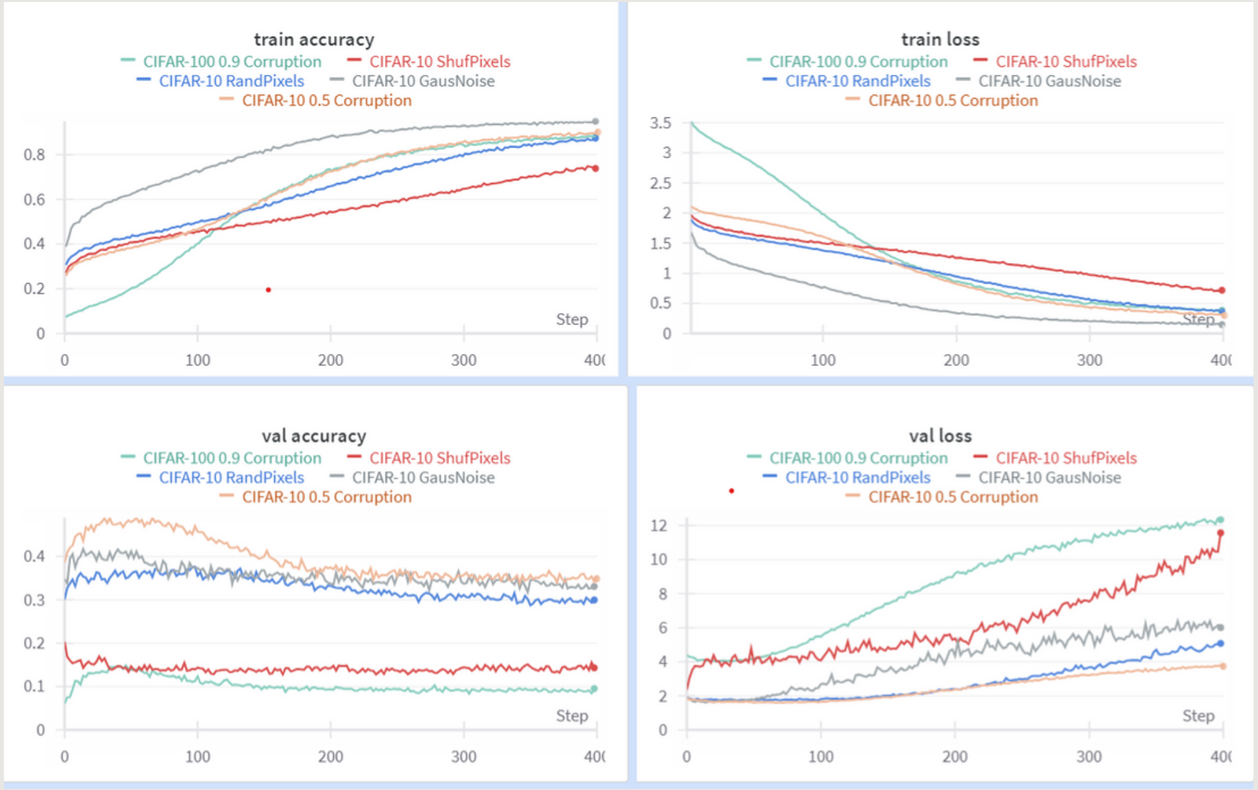
Weight decay (also known as l2-regularisation) is a technique applied to the weights of a neural network .

We minimize a loss function compromising both the primary loss function and a penalty on the L2 Norm of the weights.

$$Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^{N} w_i^2$$

Motivation for its usage comes from the traditional knowledge that we want to bias our model towards simpler functions.

Experimentation



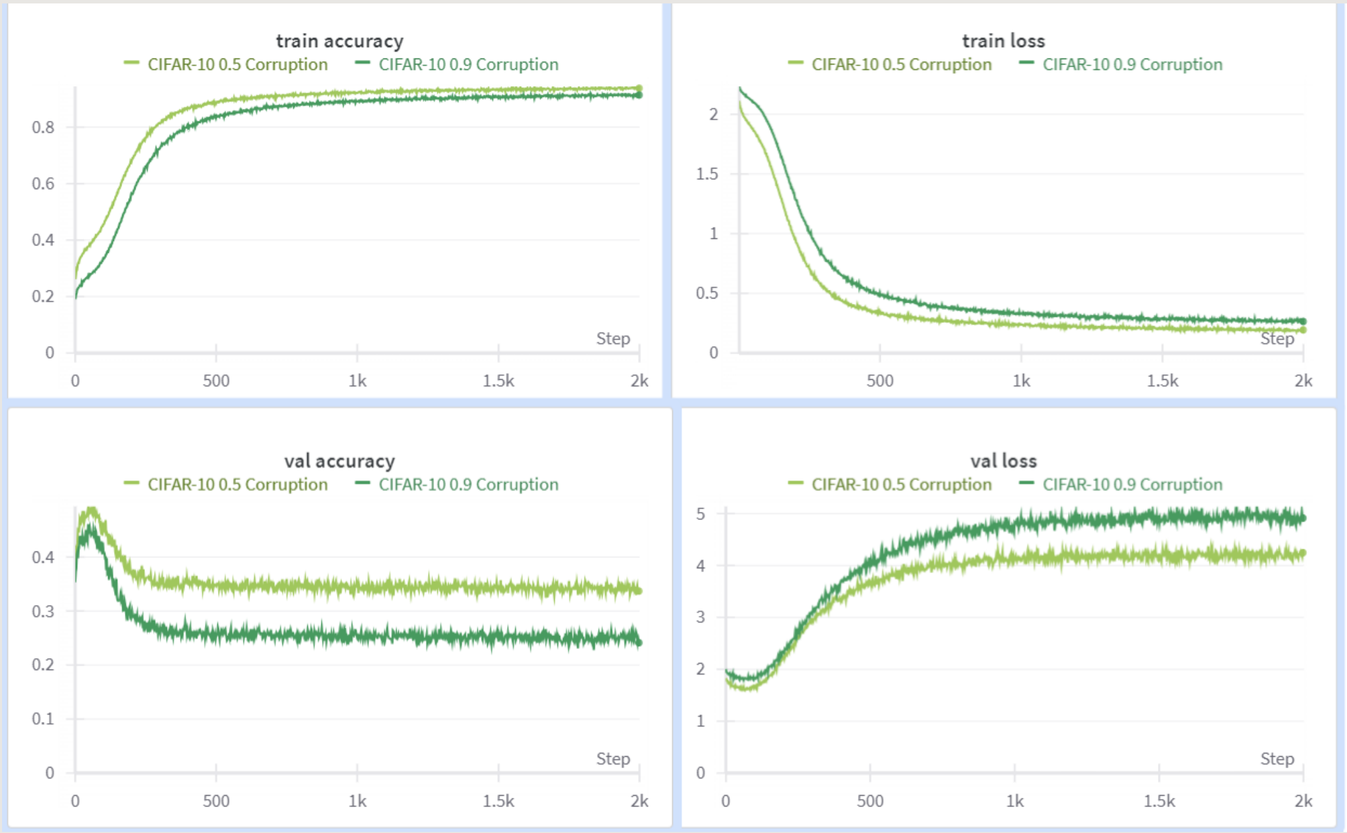
When experimented with different types of noise on CIFAR-100 and CIFAR-10 , we observe that the model still approaches 0 training loss / 100% training accuracy

Dropout + Weight Decay

In practice , deep neural network models are trained by employing both weight decay and dropout. Here are some experiments with the same.

Experimentation

Even after employing both weight decay and dropout, MLPs are still able to fit random labels.



Interpretation of results

Dropout

Dropout may improve generalization error in some cases , but doesn’t stop the model to memorize the data along with its noise.

Its working in some cases is due to the dropout noise it injects, similar to the mini-batch noise, which works as an implicit regularize and helps it generalize.

Mini-batch noise in SGD is due to the randomness it introduces during the training process which allows it escape the local minima and explore other parts of the loss landscape . Dropout, on a higher level, also “randomly” drops out some nodes which can be seen as a form of noise.

Weight Decay

Weight decay is also rarely useful as an explicit regularizer but modifies the optimization dynamics enhancing the ever-present implicit regularization of SGD via the loss stabilization mechanism.

Weight decay maintains parameters norm in a small bounded interval. The resulting projected noise-driven process induces an implicit regularization effect.

However , there is enough evidence that SGD already converges to a solution with minimum norm resulting in weight decay not able to stop training of noisy datasets.

---

*Colin Wei, Sham Kakade, Tengyu Ma - The Implicit and Explicit Regularization Effects of Dropout (2020)

**Maksym Andriushchenko, et al. - Why Do We Need Weight Decay in Modern Deep Learning?

Role of Implicit Regularization

“Implicit regularization is when machine learning algorithms, without explicitly using regularization terms, naturally produce models with simpler structures, helping prevent overfitting and enhancing generalization.”

Restricting learning process: **Early Stopping**

When updates to the parameters no longer lead to enhancements in validation, training is halted, and the most recent optimal parameters are employed.

This technique acts as a form of regularization by constraining the optimization process to a narrower range of parameter values.

Early stopping was shown to implicitly regularize on some convex learning problems.

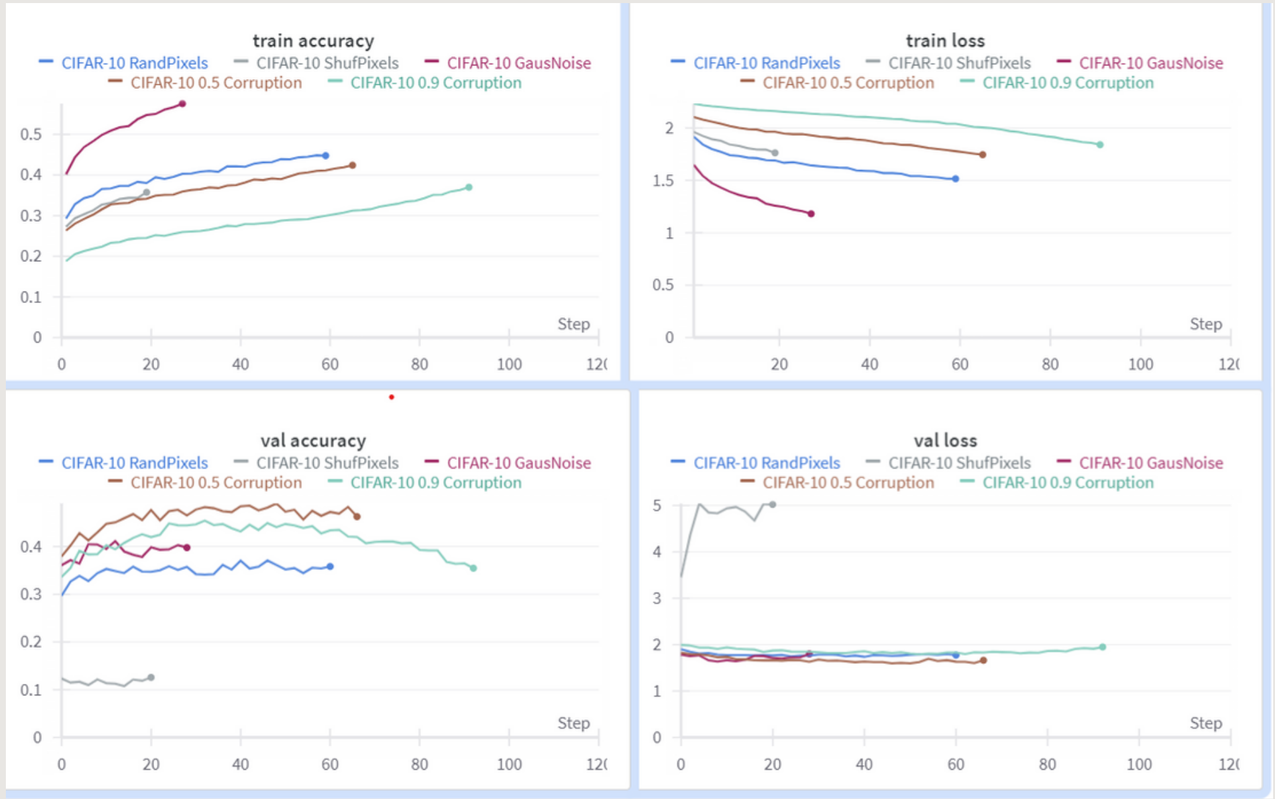
Interpretation

Early stopping is crucial for gradient descent to converge to a sparse model, a phenomenon that is called *implicit sparse regularization*.

It refers to the idea that stopping the training early can lead to simpler models with fewer parameters, which can be beneficial for preventing overfitting and improving generalization.

It implicitly penalizes overly complex models and encourages simpler models, achieving a form of sparse regularization.

Experimentation

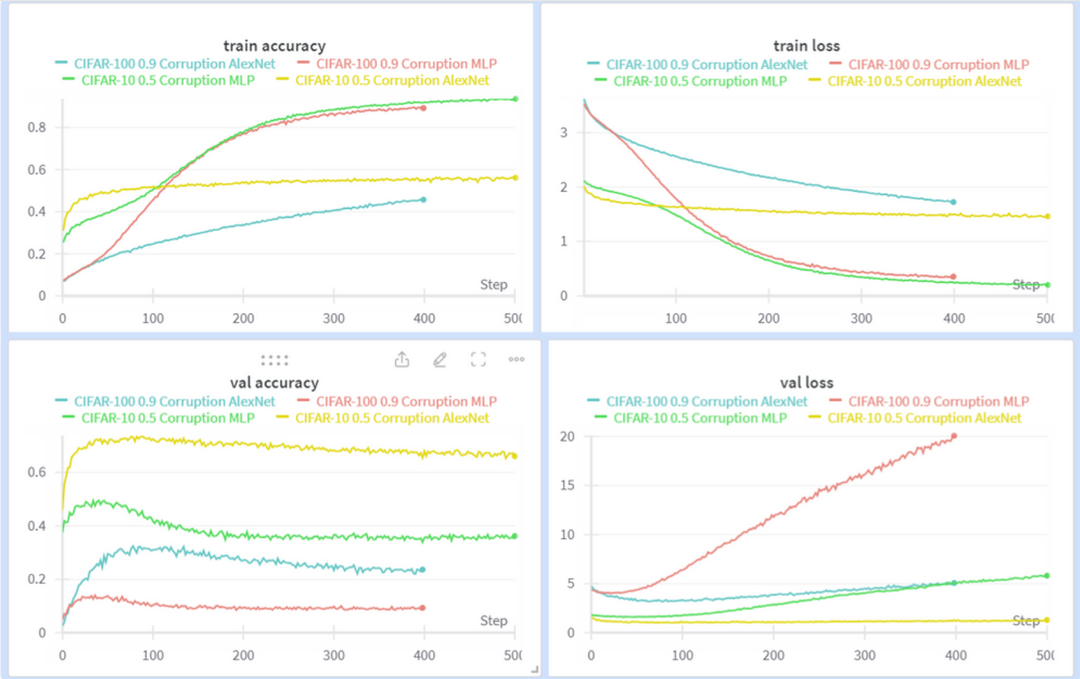


The above results show that early stopping prevents training noisy labels and noisy images by stopping the optimization process when the validation loss stabilizes

* (Yao et al., 2007; Lin et al., 2016)

Changing Model Architecture: **CNN**

CNNs preserve spatial structure in images through convolutional operations with shared kernels. The weight sharing mechanism allows these networks to capture local patterns, providing effective solutions for image recognition and classification tasks.



Random Labelling

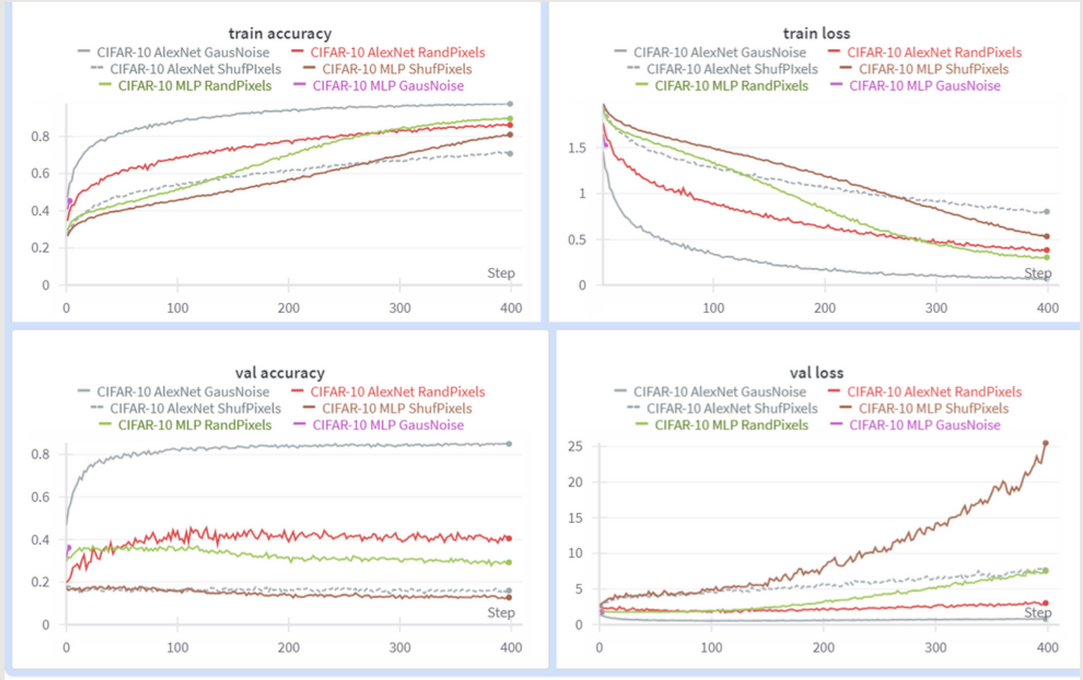


Image Noise

Interesting Observation: CNNs converge faster for random images. The possible reason being that the model gets confused when the input is correlated with each other as in the case of random labelling.

Interpretation of results

CNN's have this amazing intrinsic property of weight sharing which allows them to restrict the function hypotheses to a finite amount of functions compared to MLPs which assume that every neuron in 1 layer is related in some way to every other neuron of the next layer.

CNNs are also sparsity inducing regularizers, which induce sparsity more aggressively as the depth increases. This randomness in sparsity helps them generalize.

However, the results suggest that CNNs are able to fit random pixels , shuffled pixels and gaussian noise in the image almost similar to MLPs.

Interesting Observation: *CNNs converge faster for random images. The possible reason being that the model gets confused when the input is correlated with each other as in the case of random labelling.*

Towards a model which handles noise: IOC-NNs

“Does convexity in function help in convex optimization”*

IOC-NNs (Input Output Convex Neural Networks) impose a convexity constraint on the standard architecture. On a higher level, this is expected to minimize the hypothesis class of the model which is considered to be the traditional approach for preventing overfitting.

The convexity constraints include restricting the weights (for all but the first layer) to be non-negative and using a non-decreasing convex activation function. The final output y is convex with respect to the input x by ensuring two conditions:

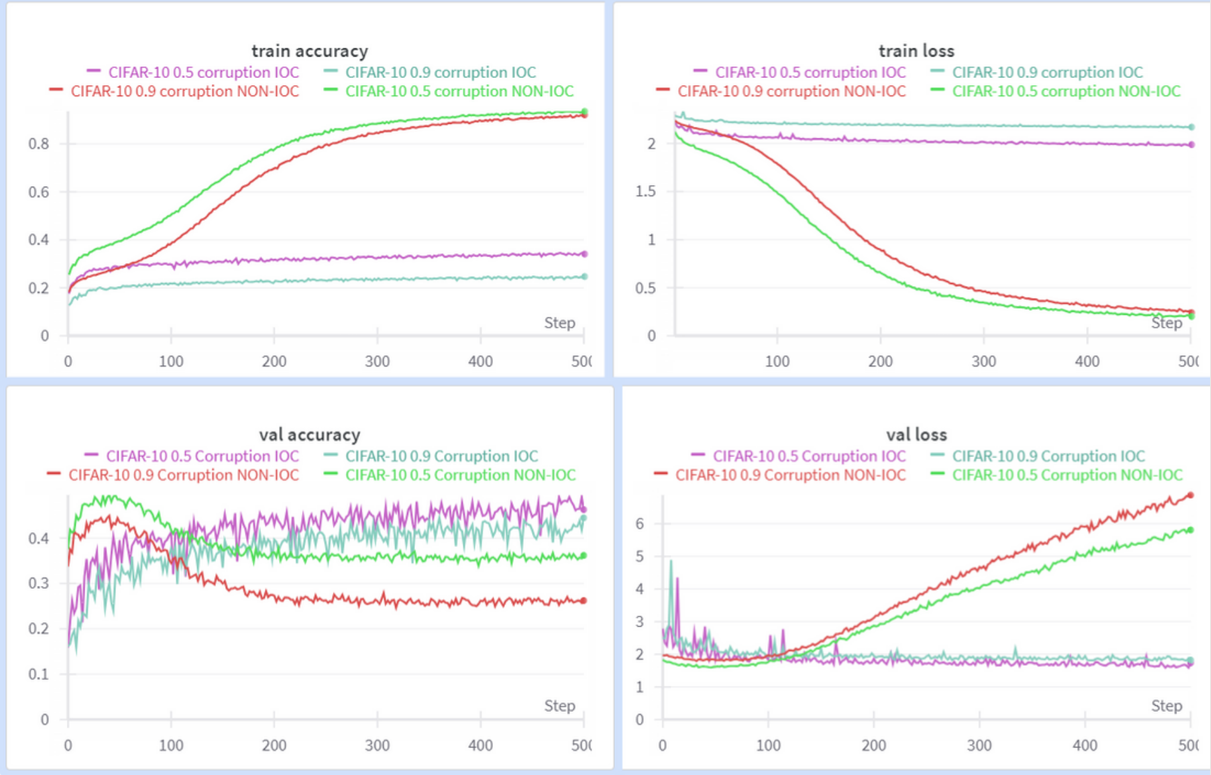
1. Weights are positive. $w_{ij} = e^{w_{ij}-5}$, for negative weights. Except for the first layer.
2. Using ELU (Exponential Linear Unit) instead of ReLU, which is a non-decreasing convex function.

---

*Vineet Gandhi, Naresh Manwani, et al. - The Curious Case of Convex Neural Networks (2020)

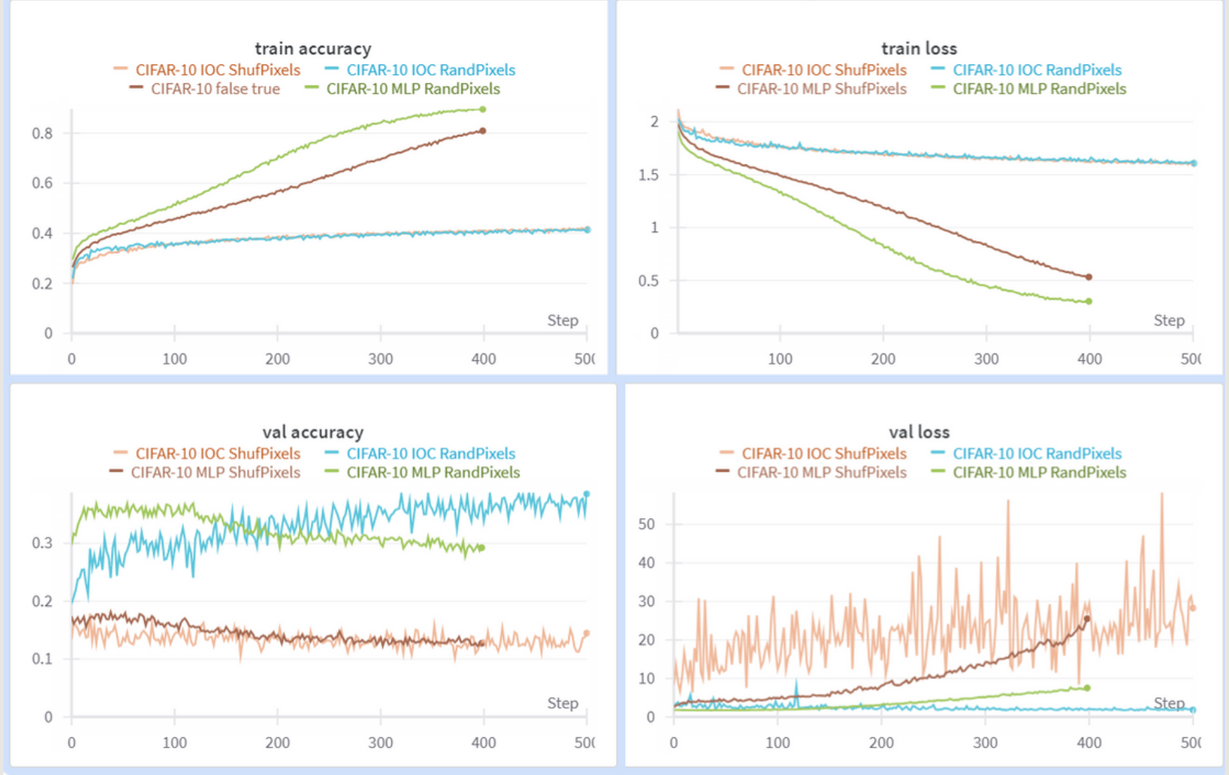
Experimentation with IOC

Corrupted Labels



IOC-NNs prevent the fitting of random labelling and the training curve flattens as compared to MLPs which approach 0 training loss.

Corrupted Images



IOC-NNs do not fit noisy images as compared to MLPs which approach 0 training loss

Interpretation of results

IOC-NNs act as self-regularisers and restrict the training process in presence of noise

The representation capacity of the resulting network reduces, and hence regularization comes in effect

However , there is still little to no explanation why these networks actually work and what kind of regularisation they impose which makes them robust to data noise.

Thus , even after a series of experiments evolving around (almost) all the traditional approches suggested to improve generalization gap and robustness of models , the question still remains :

What is it that distinguishes neural networks that generalize well from those that don’t?

Conclusion

- The paper aims at posing a very important question of formalising the methods which generalise large, over-parameterised neural network models.
- We first prove through empirical studies that these networks in effect are rich enough to memorise the entire training set, i.e give any arbitrary labelling of training data
- We then explore the regularisation techniques which are suggestive of making the model robust to data noise and observe that commonly acknowledged methods fail to generalise noisy data.
- We then describe a model (IOC-NN) and empirically show that it is able to make the model robust to training noise.
- However, all these experiments are still empirical and we have yet to discover a precise formal measure under which these enormous models are simple.

THANK YOU!