International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V73I10P108

Volume 73 Issue 10, 107-116, October 2025

Orinial Article

A Comparative Analysis of L1, L2, and L1L2 Regularization Techniques in Neural Networks for Image Classification

Deepa S[1]*, Rashmi Siddalingappa[2], Kalpana.P[1], Loveline Zeema.J[1], Vinay.M[1], Jayapriya[1], J,Suganthi.J[1], I. Priya Stella Mary[1]

[1]*Department of Computer Science, Christ University, Bengaluru, India.*
[2]*York St Jhon University, London, United Kingdom.*

[*]*Corresponding Author : sdeepa369@gmail.com*

Received: 30 May 2025 Revised: 18 September 2025 Accepted: 07 October 2025 Published: 31 October 2025

Abstract *- The research examines how L1, L2, and L1L2 weight regularization methods affect neural network performance, generalization, and sparsity using the CIFAR 10 dataset. A Convolutional Neural Network (CNN) trained with the same environment for each regularization method to evaluate test accuracy, weight sparsity, and computational speed. The study shows that L1 regularization produces sparse weights, which makes models more interpretable, and L2 regularization helps prevent overfitting while improving model generalization. The combination of L1L2 regularization enables individual image classification methods to reach test accuracy. The results indicate that the weight regularization plays a vital role in creating neural networks that are both stable and efficient. They are interpretable, and L2 regularization improves generalization and reduces overfitting. The combined L1L2 regularization achieves the balance between sparsity and performance, leading to higher test accuracy compared to individual techniques for image classification. The research results demonstrate that weight regularization stands as an essential factor for Creating Neural Networks that are robust, efficient, and interpretable, thus helping to enhance Deep Learning model performance.*

Keywords *- Weight Regularization, L1 Regularization, L2 Regularization, L1L2 Regularization, Convolutional Neural Networks (CNN), Generalization, Overfitting, Deep Learning.*

1. Introduction

Neural networks have achieved a significant amount of success in various fields. They are also responsible for the successful applications of neural networks across domains, including image classification, natural language processing, and speech recognition. The problem that is dominant in training these networks is overfitting, when a model is good at training data but is not able to generalize to unseen data.

This occurs when the model is so complex that all of the examples in the training set can be memorized instead of any underlying meaningful structure. Addressing the problem is required to realize strong and generalized neural networks, which are definitely helpful in building better neural network models. Regularization techniques help avoid overfitting in neural networks by introducing additional constraints during the training stage. Such techniques include regularization that penalizes the absolute weights (L1 regularization) and gives an increase in the weight sparsity; regularization that penalizes the squared weight values (L2) and gives an increase in the weight smoothness; and the combination of both L1 and L2 regularization that gives an increase in the mixture of sparsity and smoothness. These not only improve generalization but also improve the interpretability of the model and reduce computational costs, reducing the weight parameters. The CIFAR-10 dataset consists of 60,000 images in 10 classes; it is the benchmark dataset within the domain of image classification. CIFAR-10 is fairly small and diverse in evaluating the impact of regularization.

The use of CIFAR-10 for a side-by-side examination of how each regularization impacts model performance in terms of test accuracy, weight sparsification, and computational efficiency. This research provides a thorough review of the L1, L2, and L1L2 regularization techniques within the context of neural networks. Among the contributions of this study is a systematic analysis of the impact of L1, L2, and L1L2 regularization methods on test accuracy and generalizability performance. The examination of the computational tradeoffs between different regularization methods and their contribution helps to design more computationally efficient and generalizable neural networks.

2. Related Work

In the machine learning domain, weight regularization is one of the most critical and widely used techniques for generalizing models in an attempt to avoid unnecessarily complex solutions from potentially leading to overfitting. In the case of neural networks, the analysis of L1 and L2 has been heavily pursued. L1 regularization, where the sum of the absolute values of the weights is penalized, induces sparsity and has gained significant application in feature selection and interpretability in both linear models and neural networks. In turn, L2 regularization is also known as weight decay, which constitutes penalization of the square of the weight, leading to smooth and stable solutions with good generalization power over the unseen data. Ng (2004) demonstrated that L2 regularization is superior to L1 if the true underlying model has non-zero weights that are widely spread across features.

Conversely, L1 regularization has performed well in sparse spaces, as shown by Tibshirani (1996), whereby the Lasso method was proposed. This is why the combination of L1 with L2 regularization has been referred to as Elastic Net regularization, to benefit from both approaches. A modified hybrid method was proposed in Zou and Hastie (2005), which is a notably efficient approach for high-dimensional problems where the data are prone to noise and redundancy.

The field of Deep Learning considers regularization today as methods like dropout (Srivastava et al., 2014), batch normalization (Ioffe & Szegedy, 2015), and weight regularization through optimization strategies as more advanced. In particular, L1 and L2 regularization are still relevant as they help manage overfitting in Convolution Neural Networks (CNN) and Recurrent Neural Networks (RNN). Research suggests that these techniques provide better generalization by reducing model complexity when restricting the magnitude of the weights. There seems to be a gap in the existing literature by focusing on L1, L2, and L1L2 regularization in more comparative terms. Most studies assess these techniques in isolation or as part of a broader set of other regularization methods.

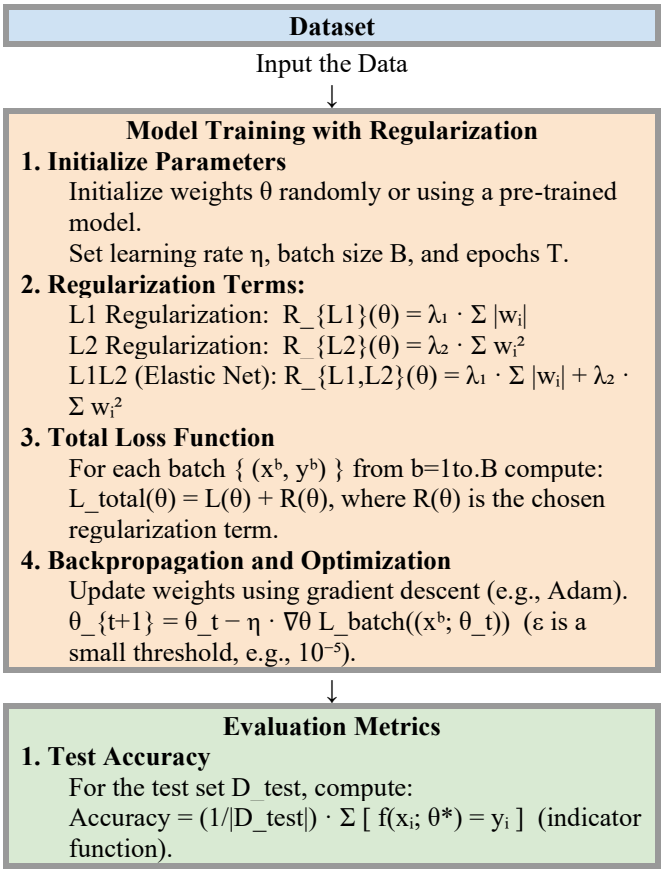
For instance, Goodfellow et al. (2016) comprehensively survey regularization approaches, but do not extensively compare L1 and L2 regularization alongside their combination. Also, although many empirical studies on CIFAR-10 have extensively studied the impact of dropout and data augmentation, the interplay of sparsity, generalization, and computational cost with L1, L2, and L1L2 regularization remains largely unexplored. This work addresses these gaps by providing a systematic comparative analysis of L1, L2, and L1L2 regularization techniques. Unlike the previous studies that emphasize absolute accuracy improvements, this work evaluates their impact on test accuracy, weight sparsity, and also computational efficiency, offering insights that bridge the gap between theoretical benefits and practical applications.

3. Methodology

The Methodology section represents the structured workflow for evaluating the different regularization techniques on Convolutional Neural Networks (CNNs), including the simple CNN and ResNet, comparing their performance based on multiple metrics.

The experiment is carried out using the CIFAR-10 dataset, which is one of the popular benchmark datasets in the field of machine learning, predominantly for evaluating the performance of image classification algorithms. It consists of 60,000 color images, and it is divided into 10 classes, with each class containing 6,000 images. The classes represent common objects such as airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Each image in the CIFAR-10 dataset is 32x32 pixels, and the images are colored (RGB format), with each pixel represented by three values corresponding to the red, green, and blue channels. The dataset is balanced, meaning that each of the 10 classes contains an equal number of images (6,000 images per class). CIFAR-10 is commonly used because of its relatively small size, making it suitable for quick experimentation, while still posing a significant challenge for neural networks, especially when aiming for generalization in image classification tasks.

The flow step methodology process is shown in Figure 1.



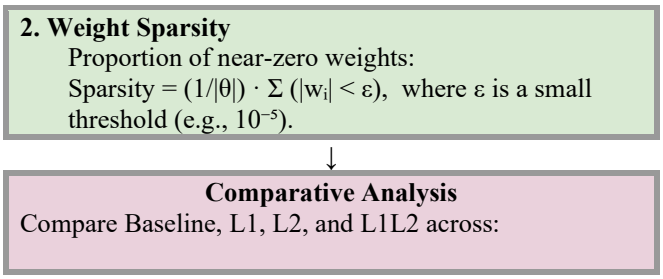


Fig. 1 Flow of methodology

3.1. Regularization Techniques

3.1.1. L1 Regularization

L1 regularization (or Lasso) operates on a simple but powerful principle: it adds a penalty equivalent to the absolute value of weight magnitudes. The mathematical formulation is in Equation (1).

$$L1=\lambda i\sum|w_i| \quad (1)$$

where _wi is the regularization hyperparameter, which controls the strength of the penalty, while wi represents the weight of each parameter. This penalty is imposed to foster some of the weights to decrease towards zero, and as a result, certain parameters are shrunk exactly to zero. The remaining set of weights is sparse with a greatly reduced number of non-zero weights, a matrix with minimally important features.

In essence, the sparsity assisted by L1 regularization brings attention to model interpretability. As sparse models focus only on a selected set of features, they become easier to analyze and understand, which makes them a favorable feature selection. In addition, L1 regularization helps high-dimensional problems where a great many features are irrelevant or redundant, which improves the efficiency of the model and captures overfitting.

3.1.2. L2 Regularization

L2 regularization, commonly referred to as weight decay, penalizes the sum of the squared values of the weights in the model. The regularization term for L2 is in Equation (2).

$$L2 = \lambda\sum w_i^2 \quad (2)$$

In this case, wi represents the weight parameters, and λ is the regularization strength. Unlike L1, the L2 regularization does not result in sparsity but instead promotes smoothness in the learned weights. By adding the squared penalty, the model is encouraged to use smaller weights, which reduces the complexity of the model and helps prevent overfitting. L2 regularization improves the generalization capability of the model by forcing the model to depend on all available features in a balanced way. It reduces more values of the weights and does not allow a single weight to dominate the learning process, which leads to more stable and reliable predictions on test data.

3.1.3. L1L2 Regularization

L1L2 regularization is also known as the Elastic Net regularization, which combines the penalties of both L1 and L2 regularization. The regularization term for L1L2 is in Equation (3).

$$L1L2= \lambda_1\sum|w_i| + \lambda_2 \sum w_i^2 \quad (3)$$

The two regularization hyperparameters λ1 and λ2 relate to the strength of the L1 and L2 penalties, respectively. This hybrid approach combines the advantages of both L1 and L2 regularization. L1 drives certain weights to zero and hence increases sparsity, whereas the L2 smoothens overall weight magnitudes. The primary advantage of L1L2 regularization is that it offers the tradeoff between sparsity and smoothness, which is beneficial for generalization in many cases. Take the scenario where the data has a large number of irrelevant features. In those cases, L1L2 regularization helps remove these features while maintaining the model's structure because of L2 smoothing. This balance can enhance the performance in high-dimensional or noisy datasets, where both methods of regularization are most beneficial. With L1 and L2 together, L1L2 regularization addresses the flexibility needed for the balance between complexity, sparsity, and generalization, resulting in model flexibility. Such features are important for the model, as they need to be interpretable because of sparsity, yet robust due to smoothness.

4. Results and Discussions

The experiments are carried out using the CIFAR-10 dataset, which contains 60,000 color images of size 32×32 pixels across 10 balanced classes. The standard preprocessing steps include normalization of pixel values to the [0,1] range and data augmentation (random horizontal flips and small shifts) to progress strength. The CNN neural network architecture With Two convolutional layers (3×3 filters, ReLU activation), Max-pooling layers for dimensionality reduction, two fully connected layers, followed by a softmax output layer. Experiments were performed with the same architecture for all regularization settings to ensure comparability.

4.1. Hyperparameters

Optimizer: Adam

Learning rate: 0.001

Batch size: 64

Regularization strengths: λ values tuned via grid search

4.2. Hardware Setup

The experiments were carried out on an NVIDIA GPU with CUDA support, 16 GB RAM, and an Intel i7 CPU. Training times were recorded to assess computational cost

differences among regularization methods. The evaluation metrics included accuracy, loss, weight sparsity (percentage of near-zero weights), and computational efficiency (time per epoch).

4.3. Baseline Model: Performance

Figure 2 indicates the training and validation accuracy trends across 12 epochs.

The training accuracy rises consistently throughout the epochs, culminating at around 85%. Validation accuracy grows significantly at the beginning, reaching a plateau of around 65-70% after a couple of epochs. Training and validation accuracy diverge, which could be a sign of overfitting. This pattern indicates that the model will memorize the training set without any provision for forcing generalization.

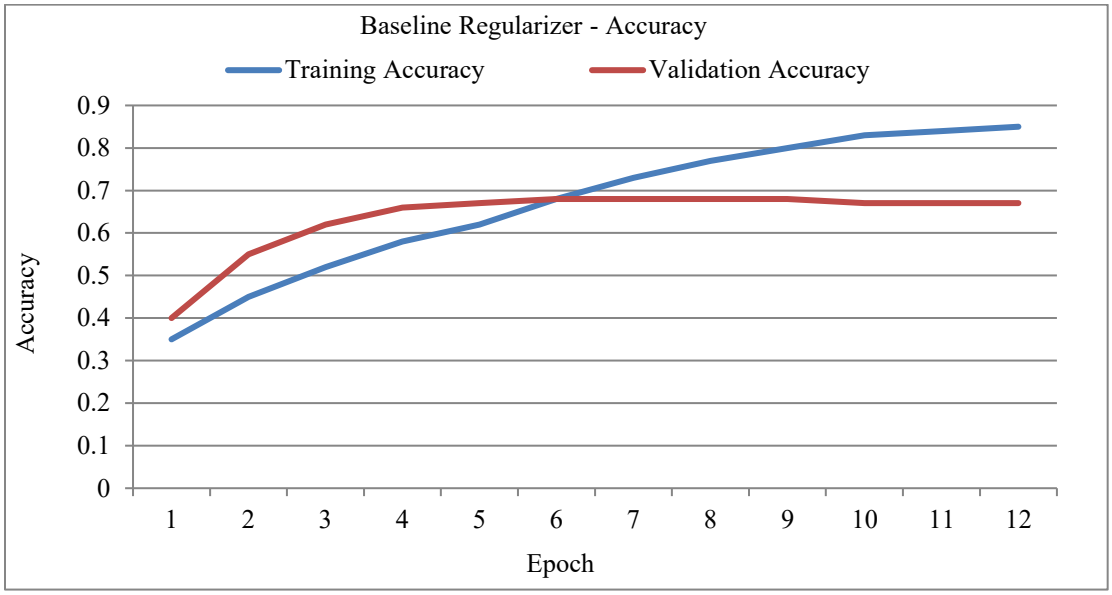


Fig. 2 Baseline regularizer - accuracy

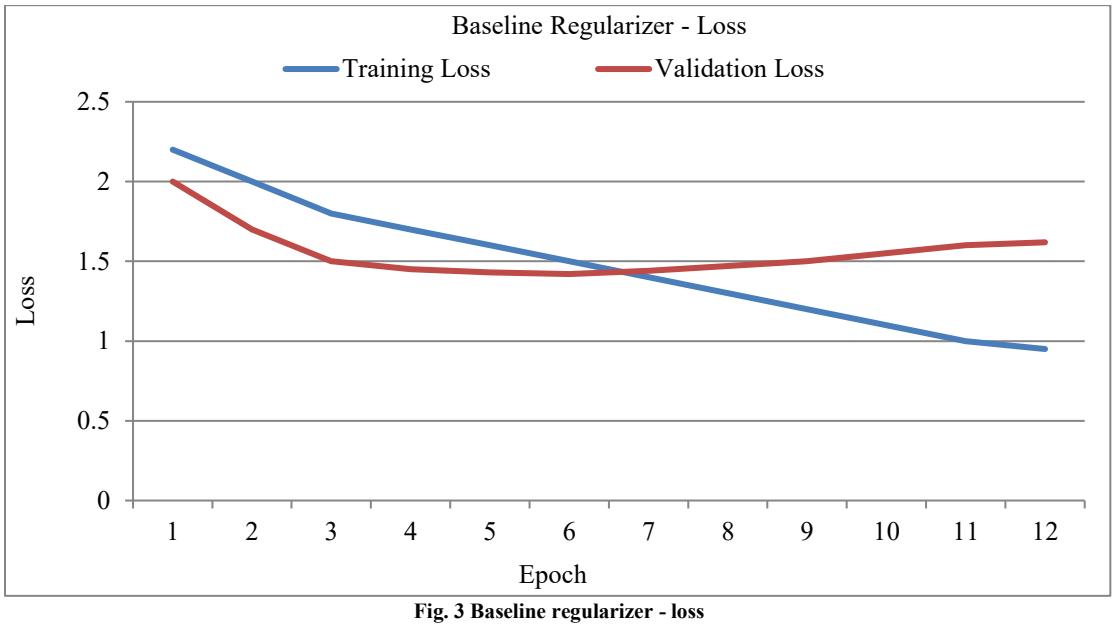


Fig. 3 Baseline regularizer - loss

Figure 3 shows the trends in loss for the training and validation sets. The loss in training goes down consistently, demonstrating efficient optimization of the model parameters on the training set. The validation loss also goes down initially, but it begins to plateau and rise a little towards the later stages of training. This is a sign of overfitting, where the model picks up noise or certain patterns in the training data that are not well-generalizable to the unseen data. The model performs well in terms of training accuracy and minimal training loss, indicating good learning. The growing

difference between the training and validation metrics highlights the poor generalization, which can be attributed to the lack of regularization. These results provide the baseline for comparing with regularized models.

Figure 4 plot shows training and validation accuracy patterns for twenty epochs for a model trained using L1 regularization, where the absolute values of the model weights are penalized in order to promote sparsity. The training accuracy begins at 44% and rises consistently to approximately 73% through the 20 epochs. This gradual improvement over the baseline (no regularization) is a manifestation of the effect of L1 regularization, which restricts the model's ability to reduce overfitting. The validation accuracy starts slightly lower than the training accuracy and rises progressively. The observed fact that validation accuracy is consistently slightly lower than training accuracy is an indication of better generalization as a result of L1 regularization. Both curves plot towards the same levels of accuracy, indicating L1 regularization benefits learning balance over training and validation sets. The improvement of training accuracy, although slower, indicates that the penalty of sparsity enforcement is confirmed by the general proximity of the training and validation accuracies to one another. This plot showcases the strength of L1 regularization in obtaining an optimal balance between training performance and generalization on unseen data.

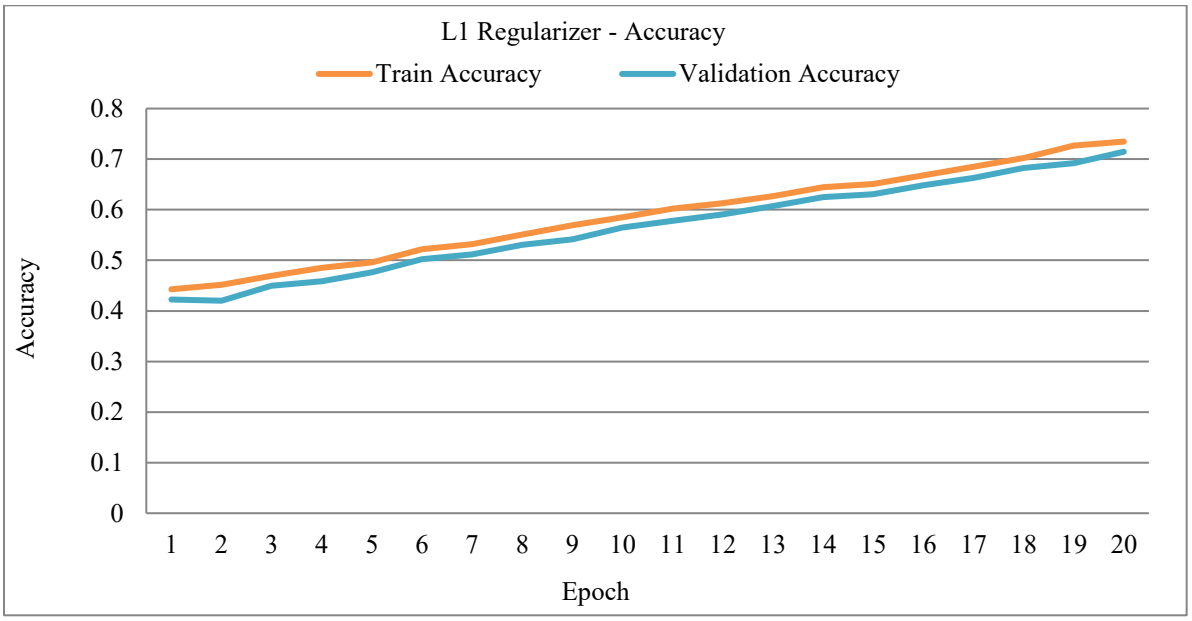


Fig. 4 L1 regularizer - accuracy

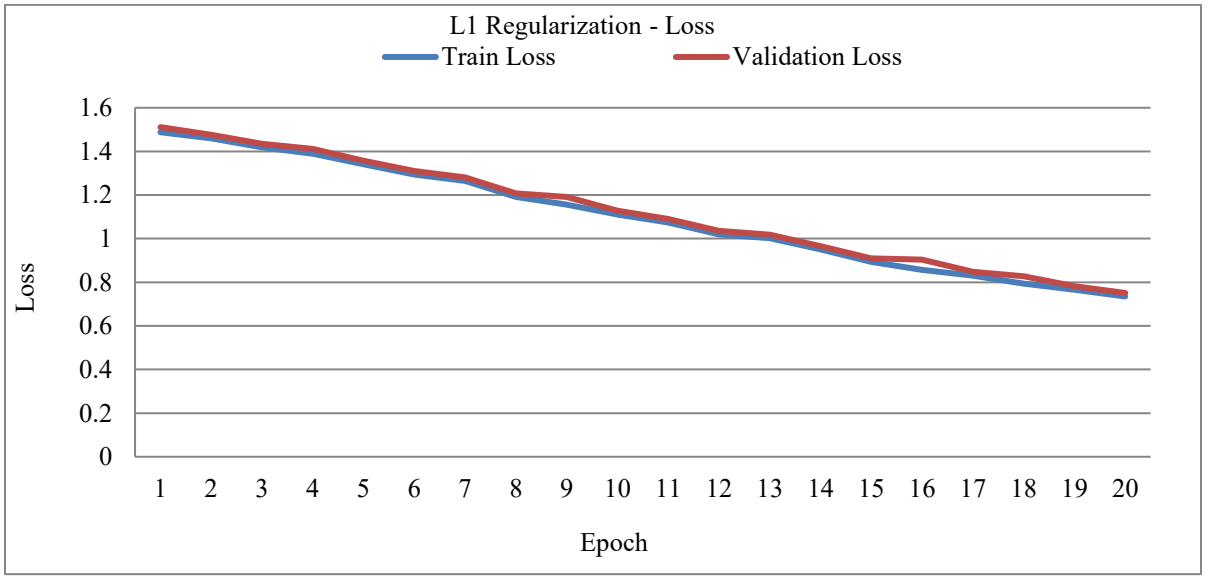


Fig. 5 L1 regularization loss

The above plot shows the loss of the model trained using L1 regularization on the training and validation set over three epochs. The x-axis is for the number of epochs, while the y-axis is for the loss values, with lower values for better model performance. The blue line follows the training loss, beginning high at around 1.5 and falling sharply to around 0.73 by epoch 20, indicating fast learning in the early epochs.

The validation loss and the training loss slowly fall close to the training loss in the last epoch. This convergence of training and validation losses indicates that the model is generalizing well without overfitting. L1 regularization penalizes the absolute weights, promoting sparsity and model interpretability. In general, the steady decline and convergence of both losses show that L1 regularization is well-balancing learning and generalization. Training accuracy begins at approximately 0.42, as shown in Figure 6, which is quite low, as would be expected for an untrained model. Training Accuracy begins slightly higher at about 0.52, possibly suggesting that the training dataset contains features that are more easily predictable by the model early on. Training accuracy rises again to about 0.7, somewhat better than validation accuracy.

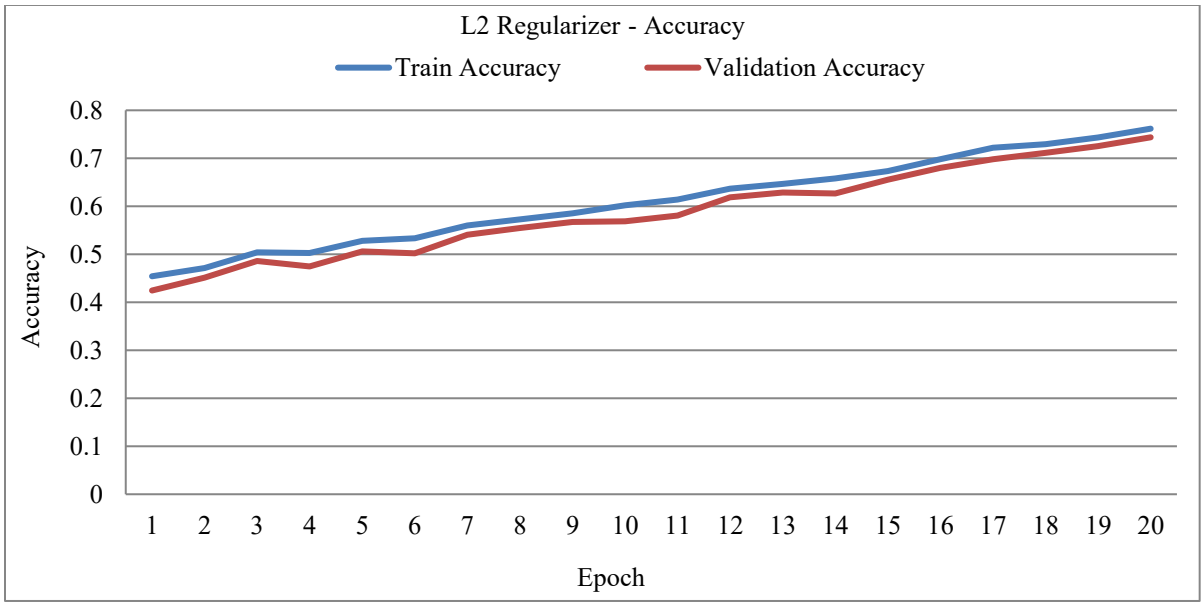


Fig. 6 L2 regularization accuracy

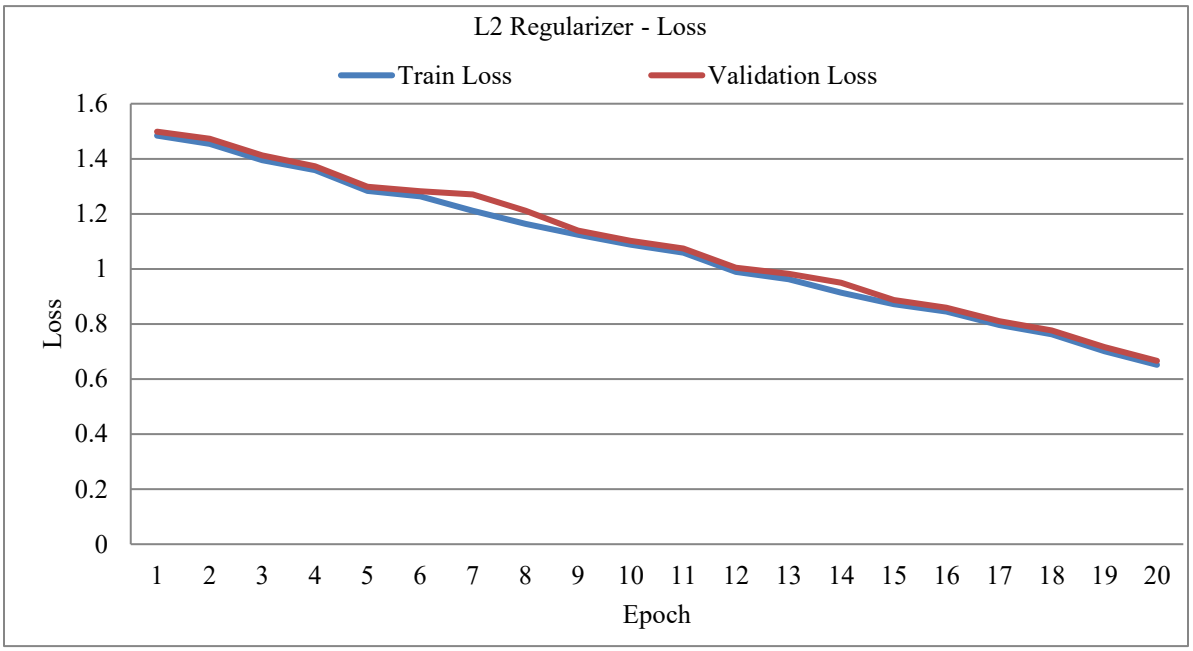


Fig. 7 L2 regularizer loss

The L2 regularizer is used to penalize large weights, which prevents overfitting. The proximity of training and validation accuracies indicates that the model is learning well without overfitting. The slightly higher training accuracy compared to validation accuracy could occur if the validation dataset is simpler or more representative than the training data. This is not a concern unless it persists significantly. Both training and validation accuracies improve steadily with each epoch, showing that the model is learning effectively without stagnation or instability. The model shows good learning dynamics with regularization, as indicated by increasing and closely matching accuracy curves. It is able to balance fitting the training data while preserving generalization to validation data. Figure 7 shows the training and validation loss curves for a model trained using L2 regularization over three epochs. The x-axis is the epochs, and the y-axis is the loss values. The training loss starts at around 1.5 at epoch zero and gradually decreases to around 0.7 by epoch 20, indicating steady improvement in how well the model can reduce error while training. The orange line, representing the validation loss, begins at a lower point around 1.5 and gradually drops to overlap closely with the training loss around 0.7 in the last epoch. This tight correlation between the training loss and validation loss underscores the fact that the model is learning well without much overfitting. L2 regularization, which penalizes the square magnitude of weights, keeps model complexity in check and adds stability, as evident from the smooth converging loss traces.

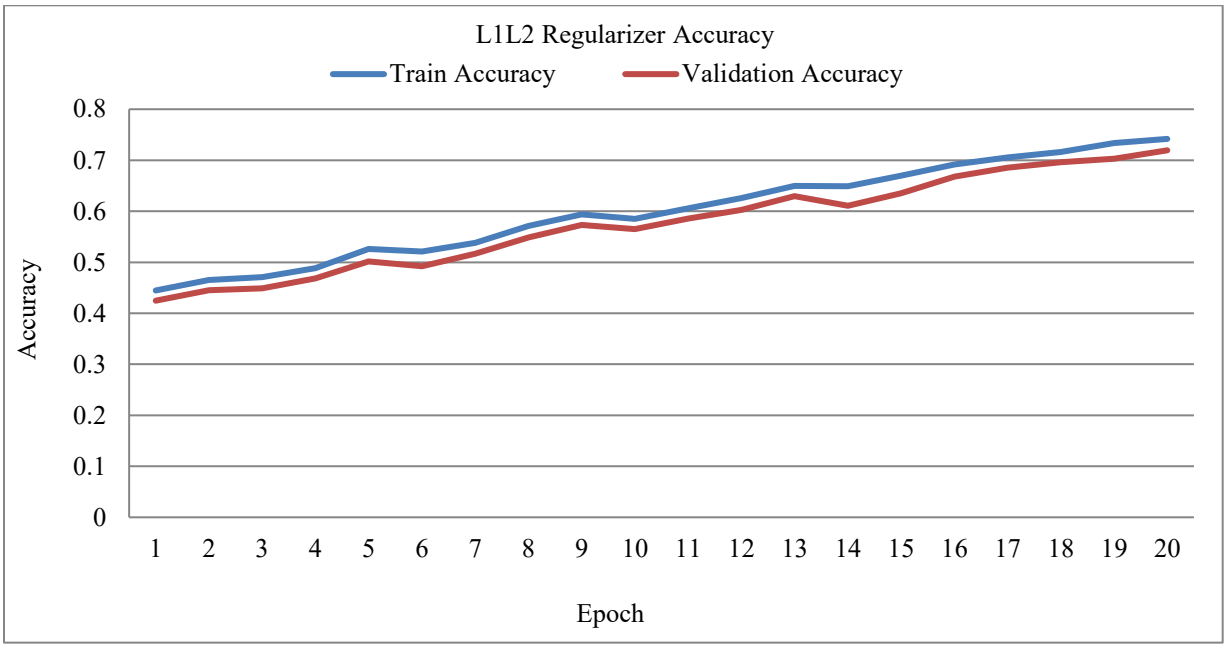


Fig. 8 L1L2 regularizer -accuracy

Training Accuracy begins at 0.45, as shown in Figure 8; Validation Accuracy begins at a lower rate than training at 0.42, which can be a function of model initialization or the distribution of the validation dataset itself. The training accuracy increases noticeably, which indicates that the model is learning from the training data. The validation accuracy also increases to 37%, demonstrating that the model is generalizing better to new data. The training accuracy further increases to 0.74, demonstrating the improvement in learning. The validation accuracy is also.

The L1L2 regularizer prevents the model from overfitting by imposing penalties on large weights. The close alignment between the training and validation accuracies across epochs suggests that the model is generalizing well without overfitting.Both training and validation accuracies improve consistently, which is a positive indicator that the regularizer and the training process are working effectively. Training Loss begins at a high point, as shown in Figure 9, typically at the start of training when the model has not yet been trained. Validation Loss begins significantly the same as the training loss, potentially reflecting the same performance on the validation set. The training loss drops precipitously. This means fast learning because the model is adapting its parameters to the training data. The validation loss also reduces slightly, indicating better generalization.

The training loss keeps falling and meets the validation loss. This convergence indicates that the model is not overfitting but generalizing very well. The L1L2 regularizer imposes penalties on large weights and prevents overfitting.

The steep decline in training loss between epochs 0 and 1 shows that the model quickly learns key patterns in the data. The gradual decrease in validation loss suggests that the model is improving its generalization without overfitting.

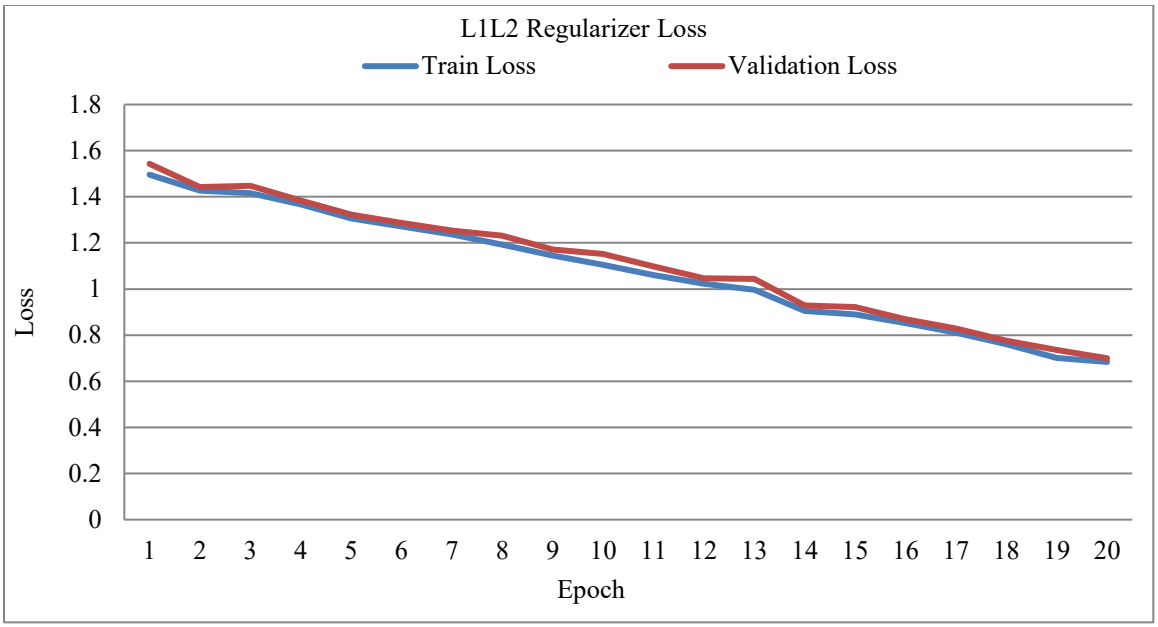


Fig. 9 L1L2 regularizer -loss

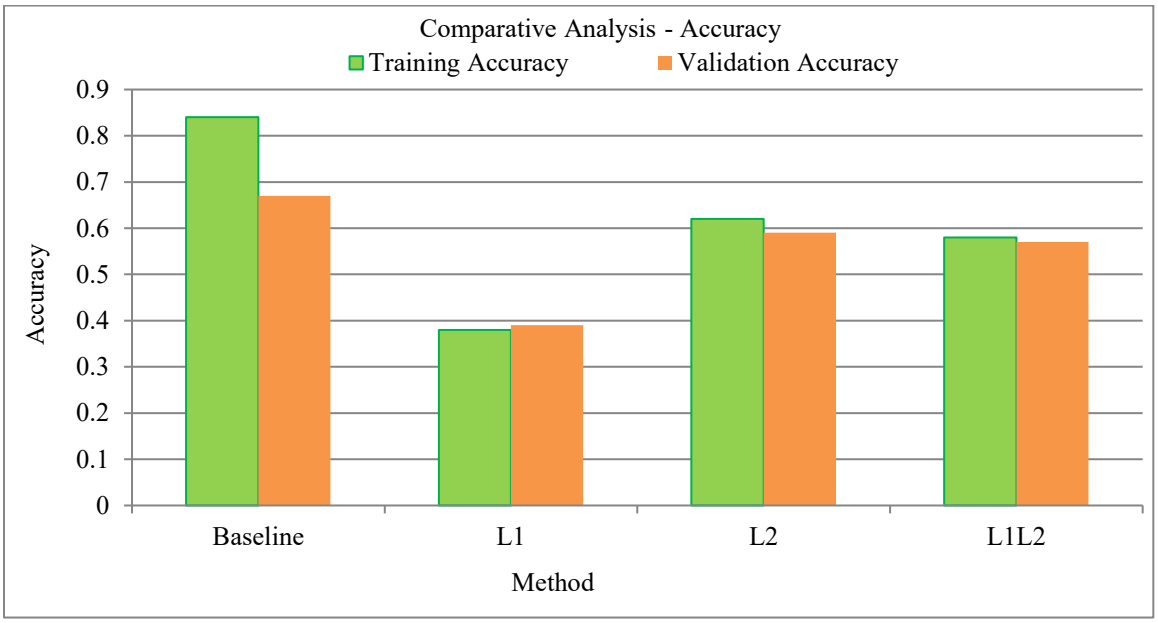


Fig. 10 Comparative analysis -accuracy

The Figure 10 graph plots the training and validation accuracy against different regularization methods: Baseline (regularization turned off), L1, L2, and L1L2 combined. Baseline model has the highest training accuracy of over 80%, while its validation accuracy is much lower, reflecting overfitting. In L1 regularization, training and validation accuracies are both considerably lower but almost the same, reflecting that although overfitting is brought under control, the overall performance takes a hit. L2 regularization, on the other hand, achieves higher and closely matched training and validation accuracies compared to L1, showing good generalization without overfitting. The L1L2 regularization brings training and validation accuracies lower than L2 but still closely matched, which shows a balance between sparsity and model complexity. In general, L2 regularization is the best approach to obtaining a good balance between training and validation accuracy, whereas the Baseline model overfits, and L1 regularization gives up some performance for simplicity. Figure 11, Training vs Validation Loss, compares the loss for models trained using various regularization methods: Baseline (no regularization), L1, L2, and L1L2 (Elastic Net). The baseline model exhibits overfitting, where training loss drops steeply while validation loss plateaus or rises. L1 regularization decreases overfitting by imposing sparsity and

results in progressively decreasing training loss but constant validation loss. L2 regularization imposes smoothness, resulting in both training and validation loss reducing progressively and also improving the generalization. L1L2 regularization combines the strengths of both L1 and L2 with the optimal tradeoff through highly correlated training as well as validation loss, indicating good generalization with minimal overfitting. This directs the need for regularization to improve the model performance, with L1L2 offering the best balanced solution.

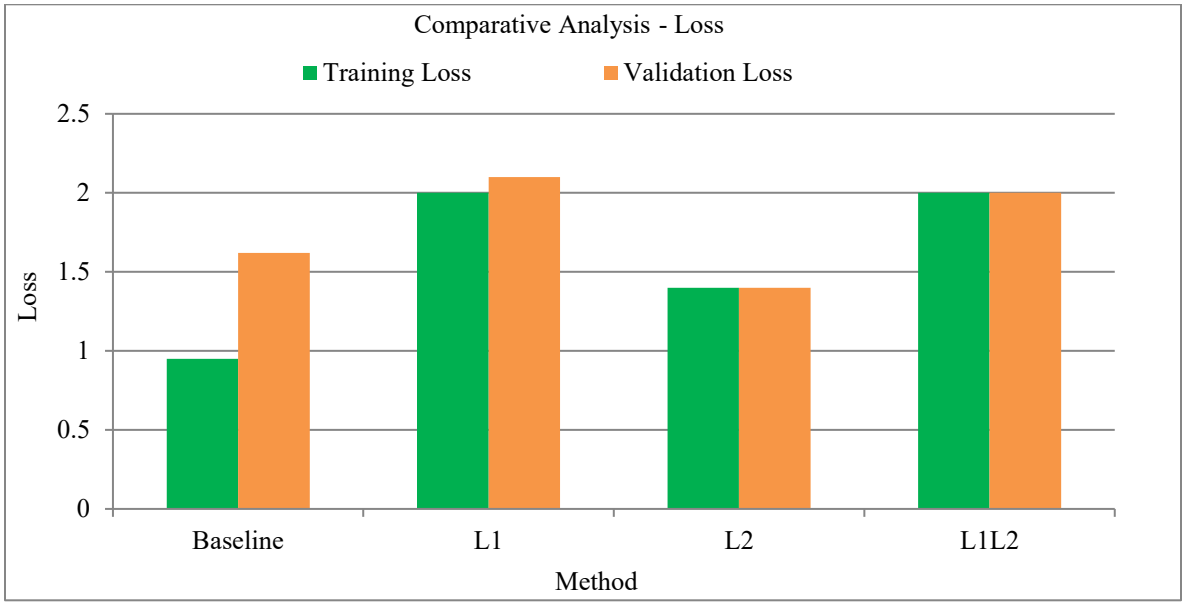


Fig. 11 Comparative analysis -loss

This section explains the experiments comparing L1, L2, and L1L2 regularization using the CIFAR-10 data, demonstrating their tradeoffs. L1 regularization encourages sparsity, making models more interpretable and also simpler, but possibly reducing accuracy by over-penalizing important features. L2 regularization smooths weight sizes, improving generalization and reducing overfitting, though not inducing sparsity, which may result in leaving models more complex.

L1L2 regularization (Elastic Net) compromises between sparsity and smoothness, yielding the best generalization, though at slightly higher computational cost due to mixed penalties. For CIFAR-10, L1 regularization is poor as it over-penalizes the features, while L2 regularization works well with the convolutional models by having the weights that are well-balanced.

L1L2 regularization is ideal for CIFAR-10 as it incorporates the best of both methods and works well with its diverse features. The biggest challenge is selecting the optimal regularization coefficients since incorrect values can lead to underfitting or overfitting. L1 regularization is susceptible to training instability due to abrupt weight changes, and introducing both L1 and L2 penalties increases computational expense, especially in larger models. The choice of regularization then depends on model and dataset requirements, with L1L2 offering the best tradeoff for CIFAR-10. The experiment results are consistent with earlier insights from Ng (2004) and Zou and Hastie (2005), but we extend their relevance to CNN-based image classification. Unlike much of the previous work that mainly focuses on accuracy, we draw attention to interpretability and efficiency as equally important factors in designing deep learning models. While modern architectures such as ResNet and DenseNet achieve higher absolute accuracy on CIFAR-10, our study offers a different perspective: it shows that classical regularization methods still play a crucial role in balancing accuracy with practical tradeoffs during neural network training.

5. Conclusion and Future Work

The study verifies the gain in performance by several regularization techniques. L1 regularization sparsifies and makes models more interpretable, but reduces accuracy as too many weights are being eliminated. L2 regularization ensures smooth weight variations that improve generalization without sacrificing accuracy. L1L2 regularization (Elastic Net) introduces the best blend, providing sparsity as well as generalizability to an optimal degree.

All regularization techniques achieve effective reduction of overfitting compared to the baseline model, with L1L2 achieving the lowest training vs validation performance gap. Regularization is most important for small datasets such as CIFAR-10, where overfitting is a big problem, and the selection of the technique depends on the model and dataset in question. For practitioners, choosing the appropriate regularization technique requires thinking about dataset size,

feature complexity, and available computational resources. L1 regularization is best suited for interpretability-intensive tasks, whereas L2 or L1L2 regularization is advised for overall performance gain. Hyperparameter optimization, e.g., grid search or random search, needs to be done in order to optimize regularization coefficients. Subsequent research can utilize these techniques on large datasets such as ImageNet to experiment with scalability and stability.

References

[1] Alex Krizhevsky, *Learning Multiple Layers of Features from Tiny Images*, University of Toronto, pp. 1-40, 2009. [Google Scholar] [Publisher Link]

[2] Andrew Y. Ng, "Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance," *ICML '04: Proceedings of the Twenty-First International Conference on Machine learning*, Banff, Alberta, Canada, pp. 1-20, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016. [Google Scholar] [Publisher Link]

[4] Hui Zou, and Trevor Hastie, "Regularization and Variable Selection via the Elastic Net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301-320, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[5] Kevin Murphy et al., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014. [Google Scholar] [Publisher Link]

[6] Kaiming He et al., "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778, 2016. [Google Scholar] [Publisher Link]

[7] Sergey Ioffe, and Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, vol. 37, pp. 448-456, 2015. [Google Scholar] [Publisher Link]

[8] Chiyuan Zhang et al., "Understanding Deep Learning Requires Rethinking Generalization," *arXiv Preprint*, pp. 1-15, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ilya Loshchilov, and Frank Hutter, "Decoupled Weight Decay Regularization," *arXiv Preprint*, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Dongyoon Han, Jiwhan Kim, and Junmo Kim, "Deep Pyramidal Residual Networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5927-5935, 2017. [Google Scholar] [Publisher Link]

[11] Karen Simonyan, and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv Preprint*, pp. 1-14, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[12] Diederik P. Kingma, and Jimmy Ba, "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations (ICLR)*, San Diego, pp. 1-15, 2015. [CrossRef] [Google Scholar] [Publisher Link]