Strategies for Improving Accuracy in Plateaued Deep Learning Models

Introduction: The Challenge of Plateauing Accuracy in Deep Learning Models

In the realm of machine learning, particularly within deep learning, a common challenge encountered during model training is the phenomenon of accuracy plateauing. This occurs when a model's performance, as measured by a metric such as accuracy on a validation dataset, ceases to improve significantly with further training, despite a continued decrease in the training loss.[1] While initially, extending the training duration might seem like a straightforward approach to enhance performance, a point is often reached where additional epochs yield diminishing returns or even lead to overfitting, where the model memorizes the training data but fails to generalize to new, unseen data.[1] To overcome this stagnation and push the boundaries of model accuracy, a more nuanced and strategic approach focusing on the network definition and the training process itself is required. This report will explore a range of advanced techniques that go beyond prolonged training, delving into methods to enhance data diversity, optimize hyperparameters, innovate network architecture, apply effective regularization, refine weight initialization, leverage advanced optimization algorithms, and strategically adjust learning rates. By addressing these critical aspects, practitioners can aim to break through accuracy plateaus and achieve state-of-the-art performance in their deep learning models.

Enhancing Data Diversity Through Augmentation

A pivotal strategy for improving the generalization and accuracy of a machine learning model, especially when its performance has plateaued, involves augmenting the training data. Data augmentation is a technique that artificially expands the size of a training dataset by creating modified versions of the existing data.[3] This process is particularly beneficial as it exposes the model to a wider range of variations, which can prevent it from overfitting to the specific nuances of the original training set and enable it to learn more robust and generalizable features.[3]

Common Image Data Augmentation Techniques and Their Application

For tasks involving image data, a variety of augmentation techniques can be employed to introduce diversity. Geometric transformations are a fundamental category, including operations such as flipping images horizontally or vertically, rotating them by small angles, scaling their size, translating them across the frame, and cropping different portions.[3] For instance, rotating an image by a few degrees can train the model to recognize the object even if it's slightly tilted in real-world scenarios.[3] If a model has become too accustomed to the specific orientations present in the training data, these geometric variations can force it to learn more invariant features.

Color space transformations offer another avenue for augmentation by adjusting the brightness, contrast, saturation, and hue of images.[3] Real-world images are captured under diverse lighting conditions, and by simulating these variations during training, the model can become more resilient to such changes. If the model's performance is sensitive to particular color distributions, these augmentations can help it generalize better to images with different lighting.

Beyond these basic techniques, more advanced methods exist, such as elastic transformations that apply non-rigid distortions to the image, and noise injection that adds random perturbations to pixel values.[3] Mixup is a technique that creates new training samples by blending two images and their corresponding labels, while Cutout involves randomly masking out rectangular regions of the image, forcing the model to learn from the remaining parts.[10] These advanced augmentations can enhance the model's robustness to various real-world challenges like partial occlusions or noisy inputs, which might be hindering further accuracy improvements.

Implementing Data Augmentation with PyTorch Transforms

Deep learning frameworks like PyTorch provide convenient tools for implementing these data augmentation techniques. The torchvision.transforms module offers a wide array of pre-built transformations that can be easily applied to image data.[3] For example, to apply random rotations and resizing to images, one can use transforms.RandomRotation and transforms.RandomResizedCrop.[3] Multiple transformations can be chained together using transforms.Compose, allowing for the creation of complex augmentation pipelines.[3]

Normalizing the data using transforms.Normalize is also crucial as it helps in speeding up the convergence of the model during training.[3] By demonstrating the practical implementation of these techniques, practitioners can readily incorporate them into their training workflows to enhance the diversity of their datasets.

Data augmentation is a powerful approach to expand the effective size of the training dataset and introduce the variability needed for a model to learn more robust features.[3] This expanded diversity makes the model less prone to overfitting the specifics of the original training data, thereby improving its ability to generalize to new, unseen examples. The choice of which augmentation techniques to apply should be carefully considered based on the nature of the data and the task at hand; for example, vertical flipping might be appropriate for some types of images but not for others like handwritten digits.[3] The effectiveness of data augmentation lies in its ability to create a training set that better reflects the real-world data distribution, leading to a model that performs more accurately on unseen data. This fundamental technique is a cornerstone of modern deep learning and often plays a crucial role in achieving state-of-the-art results, especially when the amount of original training data is limited.

Strategic Hyperparameter Optimization

Hyperparameter tuning is a critical step in maximizing the accuracy of a machine learning model, particularly when its performance has plateaued.[21] Hyperparameters are the settings of a model that are not learned from data but are set prior to the training process. The default hyperparameters of a model might not be optimal for a specific dataset and task, and systematically searching for the ideal configuration can lead to significant improvements in performance.

Exploring the Hyperparameter Landscape: Grid Search, Random Search, and Bayesian Optimization

Various methods exist for exploring the hyperparameter space. Grid search is an exhaustive technique that involves defining a set of possible values for each hyperparameter and then training and evaluating the model for every possible combination of these values.[21] This method is systematic and guarantees that the best combination within the specified search space will be found. However, it can be computationally very expensive, especially when dealing with a large number of hyperparameters or a wide range of values for each.

Random search offers a more efficient alternative, particularly in high-dimensional hyperparameter spaces.[21] Instead of trying every combination, random search selects hyperparameter combinations randomly from a defined space for a fixed number of iterations. This approach can explore a wider range of hyperparameter values compared to grid search with the same computational budget, and it is often more effective when only a few hyperparameters significantly impact the model's performance.

Bayesian optimization is a more intelligent and efficient approach that uses a probabilistic model to guide the search for the optimal hyperparameters.[21] It works by building a surrogate model of the objective function (the metric being optimized, such as validation accuracy) and uses this model to decide which hyperparameters to evaluate next. By learning from previous evaluations, Bayesian optimization balances the exploration of new hyperparameter values with the exploitation of promising regions, often finding optimal or near-optimal configurations in fewer iterations than grid or random search, especially for objective functions that are expensive to evaluate. Frameworks like Optuna provide efficient implementations of various state-of-the-art optimization algorithms, including Bayesian methods, to streamline the hyperparameter tuning process.[24]

Systematic exploration of hyperparameters can lead to the discovery of model configurations that yield higher accuracy than those obtained with default settings. The optimal hyperparameter settings are highly specific to the dataset and the model architecture, meaning there is no universal set of optimal values. By carefully tuning these settings, practitioners can significantly improve the model's ability to learn from the data and generalize to unseen examples. This process is a critical component of achieving peak performance in machine learning models.

Method	Search Strategy	Advantages	Disadvantages	When to Use

Grid Search	Exhaustive search of all combinations	Finds optimal combination within the defined space	Computationally expensive, Limited by the defined range	When the search space is small and well-defined
Random Search	Random sampling of combinations	More efficient for high-dimensional spaces	May miss the absolute best combination	When the search space is large or when only a few hyperparameters are important
Bayesian Optimization	Probabilistic model guides the search	Efficient, Balances exploration and exploitation, Fewer evaluations needed	Can be complex to implement	For expensive-to-evaluate functions or when looking for efficiency

Architectural Innovations for Improved Feature Extraction

The architecture of a neural network is fundamental to its ability to learn complex patterns from data and achieve high accuracy.[6] When a model's accuracy plateaus, it might indicate that the current architecture is a bottleneck, and exploring different network designs could be necessary to achieve further improvements.

A Review of Effective Neural Network Architectures (e.g., LeNet-5, AlexNet, VGG, ResNet)

LeNet-5 is a pioneering convolutional neural network (CNN) architecture that was highly successful for handwritten digit recognition on the MNIST dataset.[60] It consists of convolutional layers for feature extraction, pooling layers for reducing dimensionality, and fully connected layers for the final classification. LeNet-5 demonstrated the effectiveness of CNNs for tasks involving spatial data like images. For image-based tasks where the current model is not a CNN, exploring architectures like LeNet-5 or its more modern counterparts could be a beneficial starting point.

AlexNet is a deeper CNN architecture that achieved a significant breakthrough in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC).[85] It comprises eight layers, including convolutional layers, fully connected layers, and introduced key innovations such as the ReLU activation function and dropout regularization. AlexNet proved the power of deep CNNs for handling complex image classification tasks on large datasets. For problems involving more intricate image data than MNIST, considering architectures like AlexNet with increased depth and capacity might be a suitable direction.

Very Deep Convolutional Networks, or VGG, are characterized by their use of very deep stacks of convolutional layers with small 3x3 filters.[92] Architectures like VGG-16 and VGG-19

demonstrated that increasing the depth of a network with a consistent and simple structure can lead to substantial improvements in image recognition accuracy. If the current model is relatively shallow, exploring deeper architectures like VGG, which focus on stacking smaller convolutional layers to capture more complex features, could be a valuable approach.
Residual Networks, or ResNets, introduced the concept of residual connections, also known as skip connections, which allow for the training of extremely deep neural networks.[99] These connections help to address the vanishing gradient problem, which can occur in very deep networks, making it possible to train networks with hundreds or even thousands of layers. ResNet architectures have shown remarkable performance on various image recognition tasks. For very deep models or when encountering issues related to gradient flow, exploring ResNet-like architectures is highly recommended.

Leveraging Pre-trained Models with Transfer Learning

Another powerful approach to enhance accuracy, especially when dealing with limited data or when the model has plateaued, is transfer learning.[68] This technique involves using models that have been pre-trained on very large datasets, such as ImageNet, as a starting point for a new, related task. Pre-trained models have already learned a wide range of general features from these massive datasets, which can be highly beneficial even for tasks with smaller datasets.
There are two main strategies in transfer learning: feature extraction and fine-tuning.[107] Feature extraction involves using the pre-trained model to extract meaningful features from the new data and then training only a new classifier on top of these features. In this approach, the weights of the base pre-trained model are frozen, preventing them from being updated during the training on the new task. Fine-tuning, on the other hand, involves unfreezing some of the top layers of the pre-trained model and jointly training both the newly added classifier layers and the unfrozen layers of the base model on the new dataset. This allows the model to adapt the pre-learned features to the specific nuances of the new task. The choice between feature extraction and fine-tuning often depends on the size of the new dataset and its similarity to the dataset on which the model was originally trained. If the new dataset is small or very different, feature extraction might be more appropriate to avoid overfitting. If the new dataset is larger and more similar, fine-tuning can often lead to better performance by allowing the model to further specialize its learned features.

Architecture	**Key Features**	**Typical Use Cases**	**Advantages**
LeNet-5	Convolutional and pooling layers, Fully connected	Handwritten digit recognition	Simple, Effective for basic image tasks
AlexNet	Deeper CNN, ReLU activation, Dropout	Image classification on large datasets	Breakthrough performance, Introduced key innovations
VGG	Very deep CNN with small filters	Image recognition	Simple, Uniform structure, Increased

			depth improves accuracy
ResNet	Residual connections, Very deep networks	Image recognition	Overcame vanishing gradient problem, Enables training of very deep networks

Combating Overfitting with Regularization

Regularization techniques are essential tools in deep learning to prevent overfitting, a phenomenon where a model learns the training data too well, including the noise, and consequently performs poorly on unseen data.[2] Overfitting can be a significant reason for a model's accuracy to plateau.

The Role of L1 and L2 Regularization in Weight Control

L1 regularization, also known as Lasso, adds a penalty to the loss function proportional to the absolute value of the weights.[22] This encourages the weights of less important features to become zero, effectively performing feature selection and leading to a sparser model. If the model has a large number of features and some are suspected to be irrelevant, L1 regularization can be particularly useful.

L2 regularization, or Ridge regression, adds a penalty to the loss function proportional to the square of the weights.[22] This technique encourages the weights to be small, preventing them from becoming too large and thus reducing the model's complexity. L2 regularization can be beneficial when dealing with multicollinearity (high correlation between features) as it tends to spread the weight across correlated features rather than relying heavily on just one.

Dropout: Randomly Deactivating Neurons for Robustness

Dropout is a regularization technique that randomly sets a fraction of neurons in a neural network to zero during the training process.[57] This prevents the network from relying too heavily on specific neurons and encourages it to learn redundant representations, making the model more robust and improving its ability to generalize. Dropout can be particularly effective in reducing overfitting in models with a large number of parameters.

Batch Normalization: Stabilizing Activations and Accelerating Training

Batch normalization is a technique that normalizes the inputs to each layer within a mini-batch, which helps in stabilizing the training process and often allows for the use of higher learning rates.[57] By reducing the internal covariate shift (the change in the distribution of network activations due to the updates in the preceding layers' parameters), batch normalization can lead to faster convergence and more reliable training. It has also been observed to have a slight regularization effect, which can further contribute to improved generalization.

Regularization plays a crucial role in preventing overfitting, which is a common obstacle to achieving higher accuracy in deep learning models. By controlling the complexity of the model and encouraging it to learn more generalizable features, regularization techniques help bridge the gap between training and validation performance.

Optimizing the Training Process: Weight Initialization

The initial values assigned to the weights of a neural network, known as weight initialization, can significantly impact the stability and speed of the training process, as well as the final performance of the model.[166]

The Importance of Proper Initialization

Initializing all weights to zero or a constant value can lead to the symmetry problem, where all neurons in a layer learn the same features, thus hindering the model's capacity to learn complex patterns.[166] On the other hand, initializing weights with small random values can sometimes result in vanishing gradients, where the gradient signal becomes too small to effectively update the weights in the earlier layers of the network.[166] Conversely, initializing with large random values can lead to exploding gradients or saturation of activation functions, both of which can impede learning.[166] Therefore, choosing an appropriate weight initialization strategy is essential for ensuring a healthy flow of the signal (activations and gradients) through the network.

Xavier/Glorot and He Initialization Strategies

Xavier initialization, also known as Glorot initialization, is a widely used strategy, particularly for activation functions like sigmoid and tanh, which have outputs centered around zero.[166] This method scales the initial weights based on the number of inputs (fan-in) and outputs (fan-out) of each layer, aiming to maintain the variance of the activations and gradients across all layers.

He initialization, also known as Kaiming initialization, is specifically designed for ReLU (Rectified Linear Unit) and its variants, which are commonly used in modern deep learning networks.[166] ReLU activations output zero for negative inputs, which can affect the backflow of gradients. He initialization addresses this by scaling the weights based on the number of inputs to the layer, often using a scaling factor that is twice that of Xavier initialization.

Other initialization techniques, such as LeCun initialization, which focuses on preserving the variance of the forward pass, and orthogonal initialization, where weight matrices are initialized as orthogonal matrices, might be suitable for specific scenarios or network types.[167]

Choosing the right weight initialization strategy can significantly impact the training process, leading to faster convergence and potentially better final accuracy by starting the optimization in a more favorable region of the weight space.

Advanced Optimization Algorithms for Efficient Convergence

The algorithm used to update the weights of a neural network based on the gradients of the loss function, known as the optimizer, plays a crucial role in the training process and the final performance of the model.[21] When a model's accuracy plateaus, exploring advanced optimization algorithms beyond standard gradient descent might yield improvements in convergence speed and final accuracy.

Algorithm	**Key Features**	**Advantages**	**Disadvantages**
SGD with Momentum	Adds inertia to weight updates based on previous gradients	Helps escape local minima, Faster convergence than standard SGD	Requires tuning of the momentum parameter
RMSprop	Adapts learning rates for each parameter based on recent gradient magnitudes	Effective for non-stationary objectives, Resolves diminishing LR of AdaGrad	Requires tuning of learning rate and decay rate
Adam	Combines momentum and RMSprop, Adaptive learning rates for each parameter	Often faster convergence, Robust across different architectures and problems	Can sometimes generalize worse than simpler optimizers in specific scenarios

Stochastic Gradient Descent with Momentum: Adding Inertia to Learning

Stochastic Gradient Descent (SGD) with momentum is an extension of the basic SGD algorithm that adds a fraction of the previous weight update to the current update.[21] This "momentum" helps the optimizer to continue moving in a beneficial direction in the weight space and can assist in escaping shallow local minima by providing inertia to the learning process. While it can lead to faster convergence, it introduces an additional hyperparameter, the momentum factor, which needs to be tuned.

RMSprop: Adaptive Learning Rates Based on Recent Gradients

RMSprop (Root Mean Square Propagation) is an optimization algorithm that adapts the learning rate for each parameter individually based on the magnitude of the recent gradients for that parameter.[21] It maintains a moving average of the squared gradients and divides the learning rate by the square root of this average. This approach helps in resolving the issue of radically diminishing learning rates encountered in algorithms like AdaGrad and performs well in many practical applications, especially for non-stationary objectives. RMSprop requires tuning of the learning rate and a decay rate for the moving average.

Adam: Combining Momentum and Adaptive Learning Rates

Adam (Adaptive Moment Estimation) is an optimizer that combines the benefits of both momentum and RMSprop.[21] It computes adaptive learning rates for each parameter by using estimates of both the first moment (mean) and the second moment (uncentered variance) of the gradients. Adam is often considered a robust and efficient optimizer that tends to converge quickly and perform well across a wide range of neural network architectures and tasks, often requiring less hyperparameter tuning compared to other optimization algorithms. Choosing an appropriate optimization algorithm that is well-suited to the characteristics of the loss landscape can significantly impact the training process, potentially leading to faster convergence and a better final model with higher accuracy.

Refining Learning Dynamics with Learning Rate Schedules

Adjusting the learning rate during the training process, according to a predefined schedule, can be a powerful technique to improve convergence and achieve better performance when a model's accuracy has plateaued.[22]

Schedule	Description	When to Use
Step Decay	Reduces LR by a fixed factor at predefined intervals (epochs or steps)	For gradual fine-tuning after initial learning
Exponential Decay	Reduces LR exponentially over time	For stable and continuous reduction in learning rate
Cosine Annealing	Oscillates LR following a cosine function, often with warm restarts	To potentially escape local minima and improve generalization

Step Decay: Gradual Reduction at Fixed Intervals

Step decay involves reducing the learning rate by a certain factor at specific intervals, typically after a fixed number of epochs.[21] This allows for larger steps in the early stages of training when the model's weights are far from optimal, and smaller, more careful steps as training progresses and the model gets closer to convergence.

Exponential Decay: Continuous Reduction Over Time

Exponential decay reduces the learning rate exponentially over time, providing a smooth and continuous decrease throughout the training process.[21] This schedule can help in achieving stable convergence and fine-tuning the model's weights over a longer period.

Cosine Annealing: Cyclic Learning Rate Adjustments

Cosine annealing is a learning rate schedule that oscillates the learning rate following a cosine function.[26] It often includes warm restarts, where the learning rate is periodically reset to a higher value. This cyclic behavior can help the model escape local minima and potentially improve generalization.

Dynamically adjusting the learning rate can lead to faster convergence and better fine-tuning of the model's weights, potentially pushing the accuracy beyond a plateau by allowing the optimizer to adapt its step size as the training progresses.

Boosting Performance with Ensemble Methods

Ensemble methods are powerful techniques that combine the predictions of multiple machine learning models to produce a more accurate and robust prediction than any of the constituent models alone.[21] When a model's accuracy has plateaued, ensembling can be an effective way to achieve further improvements.

Bagging (Bootstrap Aggregating) involves training multiple instances of the same algorithm on different subsets of the training data, which are created by sampling with replacement.[21] The predictions from these multiple models are then aggregated, typically by averaging (for regression) or by majority voting (for classification). Bagging helps to reduce the variance of the model.

Boosting methods, such as AdaBoost and Gradient Boosting, train multiple models sequentially, where each subsequent model attempts to correct the errors made by the previous models.[21] The final prediction is a weighted combination of the predictions from all the models in the ensemble. Boosting primarily aims to reduce the bias of the model.

Stacking is another ensemble technique that trains several base models on the same dataset and then uses another model, called a meta-learner or aggregator, to learn how to best combine the predictions of the base models.[287] The base models are typically diverse and can be of different types.

Simple ensemble techniques like max voting (for classification), averaging (for regression), and weighted averaging can also be effective.[289] These methods are often straightforward to implement and can provide a quick way to potentially improve the performance of a set of trained models. Ensemble methods can often squeeze out the last few percentage points of accuracy from a model that has plateaued by leveraging the collective intelligence of multiple learners, especially if the individual models in the ensemble are diverse and make different types of errors.

Conclusion: A Holistic Approach to Breaking the Accuracy Plateau

Improving the accuracy of a deep learning model that has plateaued at 95.82% requires a multifaceted approach that goes beyond simply increasing the number of training epochs. This report has explored several key strategies, including enhancing data diversity through augmentation, strategically optimizing hyperparameters, innovating network architecture, combating overfitting with regularization, refining weight initialization, leveraging advanced optimization algorithms, and strategically adjusting learning rates. Additionally, the power of ensemble methods in boosting performance has been discussed.

Achieving significant improvements often necessitates a combination of these techniques, carefully tailored to the specific characteristics of the problem and the dataset.

Experimentation is crucial in determining the most effective strategies, and rigorous evaluation using cross-validation is essential to ensure that the observed improvements are genuine and will generalize well to unseen data. The field of deep learning model optimization is continuously advancing, and staying informed about new research and adapting strategies accordingly will be key to pushing the boundaries of model accuracy. By taking a holistic approach and thoughtfully applying these advanced techniques, practitioners can aim to break through accuracy plateaus and achieve state-of-the-art performance in their deep learning models.

Obras citadas

1. Does increasing epochs improve accuracy. : r/learnmachinelearning - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/learnmachinelearning/comments/191vm0t/does_increasing_epochs_improve_accuracy/
2. Validation Accuracy plateaued and not increasing using CNN, fecha de acceso: mayo 2, 2025, https://datascience.stackexchange.com/questions/112742/validation-accuracy-plateaued-and-not-increasing-using-cnn
3. Data Augmentation Pytorch Techniques | Restackio, fecha de acceso: mayo 2, 2025, https://www.restack.io/p/data-augmentation-knowledge-mnist-pytorch-answer-cat-ai
4. Day 10: Data Augmentation with Fashion MNIST - Akansha Saxena, fecha de acceso: mayo 2, 2025, https://akanshasaxena.com/challenge/deep-learning/day-10/
5. Data augmentation on Fashion MNIST : r/MLQuestions - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/MLQuestions/comments/jjfg8n/data_augmentation_on_fashion_mnist/
6. Common practices to mitigate accuracy plateauing at baseline? - Stack Overflow, fecha de acceso: mayo 2, 2025, https://stackoverflow.com/questions/79582733/common-practices-to-mitigate-accuracy-plateauing-at-baseline
7. Data Augmentation for Handwritten Digit Recognition - NHSJS, fecha de acceso: mayo 2, 2025, https://nhsjs.com/2025/data-augmentation-for-handwritten-digit-recognition/
8. Data Augmentation - Deep Learning - Kaggle, fecha de acceso: mayo 2, 2025, https://www.kaggle.com/code/venkatkrishnan/data-augmentation-deep-learning
9. Data augmentation | TensorFlow Core, fecha de acceso: mayo 2, 2025, https://www.tensorflow.org/tutorials/images/data_augmentation
10. A Complete Guide to Data Augmentation | DataCamp, fecha de acceso: mayo 2, 2025, https://www.datacamp.com/tutorial/complete-guide-data-augmentation
11. Overview of popular Image Augmentation packages - Kaggle, fecha de acceso: mayo 2, 2025,

https://www.kaggle.com/code/parulpandey/overview-of-popular-image-augmentation-packages
12. Image Data Augmentation for Computer Vision - viso.ai, fecha de acceso: mayo 2, 2025, https://viso.ai/computer-vision/image-data-augmentation-for-computer-vision/
13. What are the common techniques for data augmentation in images? - Milvus Blog, fecha de acceso: mayo 2, 2025, https://blog.milvus.io/ai-quick-reference/what-are-the-common-techniques-for-data-augmentation-in-images
14. 12+ Data Augmentation Techniques for Data-Efficient ML - Research AIMultiple, fecha de acceso: mayo 2, 2025, https://research.aimultiple.com/data-augmentation-techniques/
15. The Full Guide to Data Augmentation in Computer Vision - Encord, fecha de acceso: mayo 2, 2025, https://encord.com/blog/data-augmentation-guide/
16. Image Augmentation Techniques Explained - Averroes AI, fecha de acceso: mayo 2, 2025, https://averroes.ai/blog/image-augmentation-techniques-explained
17. 5 Image Data Augmentation Techniques To Mitigate Overfitting In Computer Vision, fecha de acceso: mayo 2, 2025, https://www.labellerr.com/blog/top5-image-data-augmentation-techniques-to-mitigate-overfitting-in-computer-vision/
18. 14.1. Image Augmentation — Dive into Deep Learning 1.0.3 documentation, fecha de acceso: mayo 2, 2025, http://d2l.ai/chapter_computer-vision/image-augmentation.html
19. How to choose augmentation technique : r/deeplearning - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/deeplearning/comments/qyukoc/how_to_choose_augmentation_technique/
20. Image Augmentation for Deep Learning with Keras - MachineLearningMastery.com, fecha de acceso: mayo 2, 2025, https://machinelearningmastery.com/image-augmentation-deep-learning-keras/
21. 8 Best Ways to Increase Accuracy of Machine learning model, fecha de acceso: mayo 2, 2025, https://www.analyticsvidhya.com/blog/2015/12/improve-machine-learning-results/
22. Strategies to Enhance ML Model Accuracy: A Guide - MarkovML, fecha de acceso: mayo 2, 2025, https://www.markovml.com/blog/model-accuracy
23. What is Hyperparameter Tuning? - Hyperparameter Tuning Methods ..., fecha de acceso: mayo 2, 2025, https://aws.amazon.com/what-is/hyperparameter-tuning/
24. State-of-the-Art Machine Learning Hyperparameter Optimization ..., fecha de acceso: mayo 2, 2025, https://towardsdatascience.com/state-of-the-art-machine-learning-hyperparameter-optimization-with-optuna-a315d8564de1/
25. Hyperparameters Optimization methods – ML | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/hyperparameters-optimization-methods-ml/

26. Fine-tuning Models: Hyperparameter Optimization - Encord, fecha de acceso: mayo 2, 2025, https://encord.com/blog/fine-tuning-models-hyperparameter-optimization/
27. Hyperparameter Tuning: Grid Search, Random Search, and Bayesian Optimization, fecha de acceso: mayo 2, 2025, https://keylabs.ai/blog/hyperparameter-tuning-grid-search-random-search-and-bayesian-optimization/
28. Hyperparameter tuning | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/hyperparameter-tuning/
29. Intro to Model Tuning: Grid and Random Search - Kaggle, fecha de acceso: mayo 2, 2025, https://www.kaggle.com/code/willkoehrsen/intro-to-model-tuning-grid-and-random-search
30. Hyperparameter tuning. Grid search and random search - Your Data Teacher, fecha de acceso: mayo 2, 2025, https://www.yourdatateacher.com/2021/05/19/hyperparameter-tuning-grid-search-and-random-search/
31. Tune Hyperparameters with GridSearchCV - Analytics Vidhya, fecha de acceso: mayo 2, 2025, https://www.analyticsvidhya.com/blog/2021/06/tune-hyperparameters-with-gridsearchcv/
32. Hyperparameter Optimization With Random Search and Grid Search - MachineLearningMastery.com, fecha de acceso: mayo 2, 2025, https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/
33. Hyperparameter Tuning Using Grid Search and Random Search in Python - KDnuggets, fecha de acceso: mayo 2, 2025, https://www.kdnuggets.com/2022/10/hyperparameter-tuning-grid-search-random-search-python.html
34. Hyperparameter Tuning: Examples and Top 5 Techniques, fecha de acceso: mayo 2, 2025, https://www.run.ai/guides/hyperparameter-tuning
35. Hyperparameter Optimization in Machine Learning - arXiv, fecha de acceso: mayo 2, 2025, https://arxiv.org/html/2410.22854v1
36. What is Grid Search? - Dremio, fecha de acceso: mayo 2, 2025, https://www.dremio.com/wiki/grid-search/
37. Hyperparameter optimization - Wikipedia, fecha de acceso: mayo 2, 2025, https://en.wikipedia.org/wiki/Hyperparameter_optimization
38. What is the most efficient method for hyperparameter optimization in scikit-learn?, fecha de acceso: mayo 2, 2025, https://datascience.stackexchange.com/questions/47267/what-is-the-most-efficient-method-for-hyperparameter-optimization-in-scikit-lear
39. 3.2. Tuning the hyper-parameters of an estimator - Scikit-learn, fecha de acceso: mayo 2, 2025, https://scikit-learn.org/stable/modules/grid_search.html
40. [D] Hyperparameter optimization best practices : r/MachineLearning - Reddit, fecha de acceso: mayo 2, 2025,

https://www.reddit.com/r/MachineLearning/comments/142t43v/d_hyperparameter_optimization_best_practices/
41. What Is Bayesian Hyperparameter Optimization? With Tutorial. | articles – Weights & Biases, fecha de acceso: mayo 2, 2025, https://wandb.ai/wandb_fc/articles/reports/What-Is-Bayesian-Hyperparameter-Optimization-With-Tutorial---Vmlldzo1NDQyNzcw
42. Preferred methods of Hyperparameter Optimisation? : r/learnmachinelearning - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/learnmachinelearning/comments/1bs4pcr/preferred_methods_of_hyperparameter_optimisation/
43. Random Search for Hyper-Parameter Optimization - Journal of Machine Learning Research, fecha de acceso: mayo 2, 2025, https://jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf
44. Hyper parameters tuning: Random search vs Bayesian optimization - Cross Validated, fecha de acceso: mayo 2, 2025, https://stats.stackexchange.com/questions/302891/hyper-parameters-tuning-random-search-vs-bayesian-optimization
45. What is the objective that is optimized with Random Search? - Data Science Stack Exchange, fecha de acceso: mayo 2, 2025, https://datascience.stackexchange.com/questions/73368/what-is-the-objective-that-is-optimized-with-random-search
46. [D] Random Search, Bayesian Optimization, and Hyperband and its parameters - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/MachineLearning/comments/xh6veu/d_random_search_bayesian_optimization_and/
47. Mastering Bayesian Optimization in Data Science - DataCamp, fecha de acceso: mayo 2, 2025, https://www.datacamp.com/tutorial/mastering-bayesian-optimization-in-data-science
48. Bayesian Optimization in Machine Learning | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/bayesian-optimization-in-machine-learning/
49. Deep Learning Using Bayesian Optimization - MathWorks, fecha de acceso: mayo 2, 2025, https://www.mathworks.com/help/deeplearning/ug/deep-learning-using-bayesian-optimization.html
50. Bayesian optimization - Wikipedia, fecha de acceso: mayo 2, 2025, https://en.wikipedia.org/wiki/Bayesian_optimization
51. Practical Bayesian Optimization of Machine Learning Algorithms, fecha de acceso: mayo 2, 2025, https://proceedings.neurips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf
52. Bayesian Optimization : r/datascience - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/datascience/comments/19dbh44/bayesian_optimization/

53. [1807.02811] A Tutorial on Bayesian Optimization - arXiv, fecha de acceso: mayo 2, 2025, https://arxiv.org/abs/1807.02811
54. [D] Bayesian Optimization: does it work? : r/MachineLearning - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/MachineLearning/comments/fsupfu/d_bayesian_optimization_does_it_work/
55. Bayesian Optimization (Bayes Opt): Easy explanation of popular hyperparameter tuning method - YouTube, fecha de acceso: mayo 2, 2025, https://www.youtube.com/watch?v=M-NTkxfd7-8
56. MNIST Benchmark (Image Classification) | Papers With Code, fecha de acceso: mayo 2, 2025, https://paperswithcode.com/sota/image-classification-on-mnist
57. How to correct unstable loss and accuracy during training? - Stack Overflow, fecha de acceso: mayo 2, 2025, https://stackoverflow.com/questions/55894132/how-to-correct-unstable-loss-and-accuracy-during-training
58. why does performance of machine learning models plateau after certain amount of training data, fecha de acceso: mayo 2, 2025, https://datascience.stackexchange.com/questions/51731/why-does-performance-of-machine-learning-models-plateau-after-certain-amount-of
59. How to choose CNN Architecture MNIST - Kaggle, fecha de acceso: mayo 2, 2025, https://www.kaggle.com/code/cdeotte/how-to-choose-cnn-architecture-mnist
60. Applying Convolutional Neural Network on mnist dataset - GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/applying-convolutional-neural-network-on-mnist-dataset/
61. Utilizing various CNN architectures to perform digit recognition on the MNIST Dataset. - GitHub, fecha de acceso: mayo 2, 2025, https://github.com/Jalalbaim/MNIST-Digit-recognition
62. Convolutional Neural Network on MNIST - Equinox - Patrick Kidger, fecha de acceso: mayo 2, 2025, https://docs.kidger.site/equinox/examples/mnist/
63. MNIST with Convolutional Neural Networks - Deepnote, fecha de acceso: mayo 2, 2025, https://deepnote.com/app/svpino/MNIST-with-Convolutional-Neural-Networks-a4b3c412-b802-4185-9806-02640fbba02e
64. ConvNetJS MNIST demo - CS Stanford, fecha de acceso: mayo 2, 2025, https://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html
65. Simple MNIST convnet - Keras, fecha de acceso: mayo 2, 2025, https://keras.io/examples/vision/mnist_convnet/
66. LeNet-5 - A Classic CNN Architecture - DataScienceCentral.com, fecha de acceso: mayo 2, 2025, https://www.datasciencecentral.com/lenet-5-a-classic-cnn-architecture/
67. LeNet-5 Architecture | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/lenet-5-architecture/
68. The Architecture of Lenet-5 - Analytics Vidhya, fecha de acceso: mayo 2, 2025,

https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/
69. LeNet Architecture: A Complete Guide - Kaggle, fecha de acceso: mayo 2, 2025, https://www.kaggle.com/code/blurredmachine/lenet-architecture-a-complete-guide
70. LeNet-5 from Scratch with PyTorch A Beginner's Guide - DigitalOcean, fecha de acceso: mayo 2, 2025, https://www.digitalocean.com/community/tutorials/writing-lenet5-from-scratch-in-python
71. LeNet - Wikipedia, fecha de acceso: mayo 2, 2025, https://en.wikipedia.org/wiki/LeNet
72. LeNet-5 — A complete guide - Educative.io, fecha de acceso: mayo 2, 2025, https://www.educative.io/blog/lenet-5
73. LeNet-5: Summary and Implementation - HackMD, fecha de acceso: mayo 2, 2025, https://hackmd.io/@machine-learning/S1WvJyqmI
74. LeNet-5 Implementation Using Pytorch.ipynb - GitHub, fecha de acceso: mayo 2, 2025, https://github.com/lychengrex/LeNet-5-Implementation-Using-Pytorch/blob/master/LeNet-5%20Implementation%20Using%20Pytorch.ipynb
75. Implementing Yann LeCun's LeNet-5 in PyTorch | Towards Data Science, fecha de acceso: mayo 2, 2025, https://towardsdatascience.com/implementing-yann-lecuns-lenet-5-in-pytorch-5e05a0911320/
76. MNIST classification using LeNet on Pytorch - Kaggle, fecha de acceso: mayo 2, 2025, https://www.kaggle.com/code/yogeshrampariya/mnist-classification-using-lenet-on-pytorch
77. The simplest implementation of LeNet5 with mnist in PyTorch. Accuracy: ~99% - GitHub, fecha de acceso: mayo 2, 2025, https://github.com/ChawDoe/LeNet5-MNIST-PyTorch
78. FashionMNIST classification using LeNet-5 architecture - John Bosco, fecha de acceso: mayo 2, 2025, https://boscoj2008.github.io/LeNet-5/
79. LeNet5 based CNN with 98.94% using PyTorch - Kaggle, fecha de acceso: mayo 2, 2025, https://www.kaggle.com/code/ankschoubey/lenet5-based-cnn-with-98-94-using-pytorch
80. Handwritten Digit Recognition with LeNet5 Model in PyTorch - Machine Learning Mastery, fecha de acceso: mayo 2, 2025, https://machinelearningmastery.com/handwritten-digit-recognition-with-lenet5-model-in-pytorch/
81. 2 - LeNet.ipynb - Colab, fecha de acceso: mayo 2, 2025, https://colab.research.google.com/github/bentrevett/pytorch-image-classification/blob/master/2_lenet.ipynb
82. [D] Conv Net Question: Understanding LeNet-5 w/ MNIST : r/MachineLearning - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/MachineLearning/comments/eoyve4/d_conv_net_questi

on_understanding_lenet5_w_mnist/
83. The Convolutional Neural Network - Theory and Implementation of LeNet-5 and AlexNet, fecha de acceso: mayo 2, 2025, https://pabloinsente.github.io/the-convolutional-network
84. 7.6. Convolutional Neural Networks (LeNet) - Dive into Deep Learning, fecha de acceso: mayo 2, 2025, http://d2l.ai/chapter_convolutional-neural-networks/lenet.html
85. A Review of Popular Deep Learning Architectures: AlexNet, VGG16, and GoogleNet, fecha de acceso: mayo 2, 2025, https://www.digitalocean.com/community/tutorials/popular-deep-learning-architectures-alexnet-vgg-googlenet
86. AlexNet Explained | Papers With Code, fecha de acceso: mayo 2, 2025, https://paperswithcode.com/method/alexnet
87. AlexNet - Wikipedia, fecha de acceso: mayo 2, 2025, https://en.wikipedia.org/wiki/AlexNet
88. Writing AlexNet from Scratch in PyTorch - DigitalOcean, fecha de acceso: mayo 2, 2025, https://www.digitalocean.com/community/tutorials/alexnet-pytorch
89. AlexNet: A Revolutionary Deep Learning Architecture - viso.ai, fecha de acceso: mayo 2, 2025, https://viso.ai/deep-learning/alexnet/
90. Introduction to The Architecture of Alexnet - Analytics Vidhya, fecha de acceso: mayo 2, 2025, https://www.analyticsvidhya.com/blog/2021/03/introduction-to-the-architecture-of-alexnet/
91. Difference between AlexNet and GoogleNet - GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/difference-between-alexnet-and-googlenet/
92. Writing VGG from Scratch in PyTorch - DigitalOcean, fecha de acceso: mayo 2, 2025, https://www.digitalocean.com/community/tutorials/vgg-from-scratch-pytorch
93. What is Visual Geometry Group (VGG)? - Great Learning, fecha de acceso: mayo 2, 2025, https://www.mygreatlearning.com/blog/visual-geometry-group/
94. VGGNet - Wikipedia, fecha de acceso: mayo 2, 2025, https://en.wikipedia.org/wiki/VGGNet
95. VGG-16 | CNN model - GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/vgg-16-cnn-model/
96. VGG Explained - Papers With Code, fecha de acceso: mayo 2, 2025, https://paperswithcode.com/method/vgg
97. VGG-Net Architecture Explained - GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/vgg-net-architecture-explained/
98. VGGNet-16 Architecture: A Complete Guide - Kaggle, fecha de acceso: mayo 2, 2025, https://www.kaggle.com/code/blurredmachine/vggnet-16-architecture-a-complete-guide
99. PyTorch ResNet: The Basics and a Quick Tutorial, fecha de acceso: mayo 2, 2025, https://www.run.ai/guides/deep-learning-for-computer-vision/pytorch-resnet

100. Residual neural network - Wikipedia, fecha de acceso: mayo 2, 2025, https://en.wikipedia.org/wiki/Residual_neural_network
101. Deep Residual Learning for Image Recognition (ResNet Explained) - Analytics Vidhya, fecha de acceso: mayo 2, 2025, https://www.analyticsvidhya.com/blog/2023/02/deep-residual-learning-for-image-recognition-resnet-explained/
102. ResNet Explained - Residual Network - Papers With Code, fecha de acceso: mayo 2, 2025, https://paperswithcode.com/method/resnet
103. Deep Learning Architectures Explained: ResNet, InceptionV3, SqueezeNet | DigitalOcean, fecha de acceso: mayo 2, 2025, https://www.digitalocean.com/community/tutorials/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet
104. Residual Networks (ResNet) – Deep Learning - GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/
105. Resnet-34 Architecture. | Download Scientific Diagram - ResearchGate, fecha de acceso: mayo 2, 2025, https://www.researchgate.net/figure/Resnet-34-Architecture_fig1_354122133
106. conv_nets.pptx - SlideShare, fecha de acceso: mayo 2, 2025, https://www.slideshare.net/slideshow/convnetspptx/257941498
107. Transfer learning and fine-tuning | TensorFlow Core, fecha de acceso: mayo 2, 2025, https://www.tensorflow.org/tutorials/images/transfer_learning
108. Transfer Learning Guide: A Practical Tutorial With Examples for Images and Text in Keras, fecha de acceso: mayo 2, 2025, https://neptune.ai/blog/transfer-learning-guide-examples-for-images-and-text-in-keras
109. A Transfer Learning Evaluation of Deep Neural Networks for Image Classification - MDPI, fecha de acceso: mayo 2, 2025, https://www.mdpi.com/2504-4990/4/1/2
110. Transfer Learning for Image Classification - Kaggle, fecha de acceso: mayo 2, 2025, https://www.kaggle.com/code/kmkarakaya/transfer-learning-for-image-classification
111. Multiclass image classification using Transfer learning | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/multiclass-image-classification-using-transfer-learning/
112. Transfer Learning for Computer Vision Tutorial - PyTorch, fecha de acceso: mayo 2, 2025, https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html
113. Top 4 Pre-Trained Models for Image Classification with Python Code - Analytics Vidhya, fecha de acceso: mayo 2, 2025, https://www.analyticsvidhya.com/blog/2020/08/top-4-pre-trained-models-for-image-classification-with-python-code/
114. Image-Recognition-with-Deep-Learning/Image Classification with Transfer

Learning.md at master - GitHub, fecha de acceso: mayo 2, 2025, https://github.com/hbhasin/Image-Recognition-with-Deep-Learning/blob/master/Image%20Classification%20with%20Transfer%20Learning.md
115. Transfer learning for TensorFlow image classification models in Amazon SageMaker - AWS, fecha de acceso: mayo 2, 2025, https://aws.amazon.com/blogs/machine-learning/transfer-learning-for-tensorflow-image-classification-models-in-amazon-sagemaker/
116. Deep Transfer Learning Using Real-World Image Features for Medical Image Classification, with a Case Study on Pneumonia X-ray Images - PMC, fecha de acceso: mayo 2, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC11048359/
117. What Is Regularization? | IBM, fecha de acceso: mayo 2, 2025, https://www.ibm.com/think/topics/regularization
118. Regularization in Neural Networks | Pinecone, fecha de acceso: mayo 2, 2025, https://www.pinecone.io/learn/regularization-in-neural-networks/
119. Regularizing neural networks - DeepLearning.AI, fecha de acceso: mayo 2, 2025, https://www.deeplearning.ai/ai-notes/regularization/
120. Regularization in Deep Learning with Python code - Analytics Vidhya, fecha de acceso: mayo 2, 2025, https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/
121. Regularization Methods for Neural Networks - Sebastian Raschka, fecha de acceso: mayo 2, 2025, https://sebastianraschka.com/pdf/lecture-notes/stat453ss21/L10_regularization__slides.pdf
122. Does regularization help neural networks ignore irrelevant features or simply reduce overfitting to the data? - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/learnmachinelearning/comments/199635c/does_regularization_help_neural_networks_ignore/
123. Understanding Neural Network Regularization and Key Regularization Techniques - Zilliz, fecha de acceso: mayo 2, 2025, https://zilliz.com/learn/understanding-regularization-in-nueral-networks
124. Regularization techniques for training deep neural networks | AI Summer, fecha de acceso: mayo 2, 2025, https://theaisummer.com/regularization/
125. List here all the regularization techniques you know for Deep Learning models - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/deeplearning/comments/17eiu9p/list_here_all_the_regularization_techniques_you/
126. Regularization in Machine Learning | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/regularization-in-machine-learning/
127. Overfitting: L2 regularization | Machine Learning - Google for Developers, fecha de acceso: mayo 2, 2025, https://developers.google.com/machine-learning/crash-course/overfitting/regularization
128. how-to-use-l1-l2-and-elastic-net-regularization-with-pytorch.md - GitHub, fecha de acceso: mayo 2, 2025,

https://github.com/christianversloot/machine-learning-articles/blob/main/how-to-use-l1-l2-and-elastic-net-regularization-with-pytorch.md
129. L1 vs L2 regularization. Which is "better"? : r/learnmachinelearning - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/learnmachinelearning/comments/1eqp6bc/l1_vs_l2_regularization_which_is_better/
130. Dropout Regularization in Deep Learning - Analytics Vidhya, fecha de acceso: mayo 2, 2025, https://www.analyticsvidhya.com/blog/2022/08/dropout-regularization-in-deep-learning/
131. Dropout Regularization in Deep Learning | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/dropout-regularization-in-deep-learning/
132. Dilution (neural networks) - Wikipedia, fecha de acceso: mayo 2, 2025, https://en.wikipedia.org/wiki/Dilution_(neural_networks)
133. Overfitting Prevention with Regularization and Dropout | CodeSignal Learn, fecha de acceso: mayo 2, 2025, https://codesignal.com/learn/courses/pytorch-techniques-for-model-optimization/lessons/overfitting-prevention-with-regularization-and-dropout
134. neural networks - How to explain dropout regularization in simple terms? - Cross Validated, fecha de acceso: mayo 2, 2025, https://stats.stackexchange.com/questions/241645/how-to-explain-dropout-regularization-in-simple-terms
135. Using Dropout Regularization in PyTorch Models - MachineLearningMastery.com, fecha de acceso: mayo 2, 2025, https://machinelearningmastery.com/using-dropout-regularization-in-pytorch-models/
136. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, fecha de acceso: mayo 2, 2025, https://jmlr.org/papers/v15/srivastava14a.html
137. A Gentle Introduction to Dropout for Regularizing Deep Neural Networks - MachineLearningMastery.com, fecha de acceso: mayo 2, 2025, https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/
138. Regularization in Deep Learning: L1, L2 & Dropout - E2E Networks, fecha de acceso: mayo 2, 2025, https://www.e2enetworks.com/blog/regularization-in-deep-learning-l1-l2-dropout
139. L1 and L2 Regularization Methods, Explained - Built In, fecha de acceso: mayo 2, 2025, https://builtin.com/data-science/l2-regularization
140. L1 And L2 Regularization Explained, When To Use Them & Practical How To Examples, fecha de acceso: mayo 2, 2025, https://spotintelligence.com/2023/05/26/l1-l2-regularization/
141. Regularization (mathematics) - Wikipedia, fecha de acceso: mayo 2, 2025, https://en.wikipedia.org/wiki/Regularization_(mathematics)
142. Understanding L1 and L2 regularization: techniques for optimized model

training - Wandb, fecha de acceso: mayo 2, 2025, https://wandb.ai/mostafaibrahim17/ml-articles/reports/Understanding-L1-and-L2-regularization-techniques-for-optimized-model-training--Vmlldzo3NzYwNTM5
143. L1/L2 Regularization in PyTorch | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/l1l2-regularization-in-pytorch/
144. Understanding l1 and l2 Regularization - Towards Data Science, fecha de acceso: mayo 2, 2025, https://towardsdatascience.com/understanding-l1-and-l2-regularization-93918a5ac8d0/
145. [D] How does L1 regularization perform feature selection? - Seeking an intuitive explanation using polynomial models : r/MachineLearning - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/MachineLearning/comments/1j8gvlh/d_how_does_l1_regularization_perform_feature/
146. How is L1 Regularization able to drive a coefficient to zero? - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/learnmachinelearning/comments/1774ndz/how_is_l1_regularization_able_to_drive_a/
147. Parameter choice rules for L1 regularization? - Cross Validated - Stack Exchange, fecha de acceso: mayo 2, 2025, https://stats.stackexchange.com/questions/599830/parameter-choice-rules-for-l1-regularization
148. How to add a L2 regularization term in my loss function - autograd - PyTorch Forums, fecha de acceso: mayo 2, 2025, https://discuss.pytorch.org/t/how-to-add-a-l2-regularization-term-in-my-loss-function/17411
149. Dropout Regularization Using PyTorch: A Hands-On Guide - DataCamp, fecha de acceso: mayo 2, 2025, https://www.datacamp.com/tutorial/dropout-regularization-using-pytorch-guide
150. Dropout in Neural Networks | Dremio, fecha de acceso: mayo 2, 2025, https://www.dremio.com/wiki/dropout-in-neural-networks/
151. Dropout in Neural Networks | Towards Data Science, fecha de acceso: mayo 2, 2025, https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9/
152. Hybrid convolutional neural networks-support vector machine classifier with dropout for Javanese character recognition - SlideShare, fecha de acceso: mayo 2, 2025, https://www.slideshare.net/slideshow/hybrid-convolutional-neural-networkssupport-vector-machine-classifier-with-dropout-for-javanese-character-recognition/255830883
153. Forward only counterpropagation architecture. | Download Scientific Diagram - ResearchGate, fecha de acceso: mayo 2, 2025, https://www.researchgate.net/figure/Forward-only-counterpropagation-architecture_fig2_281564605
154. Batch normalization - Wikipedia, fecha de acceso: mayo 2, 2025,

https://en.wikipedia.org/wiki/Batch_normalization
155. Understanding the Impact of Batch Normalization on CNNs - TiDB, fecha de acceso: mayo 2, 2025, https://www.pingcap.com/article/understanding-the-impact-of-batch-normalization-on-cnns/
156. 8.5. Batch Normalization — Dive into Deep Learning 1.0.3 documentation, fecha de acceso: mayo 2, 2025, http://d2l.ai/chapter_convolutional-modern/batch-norm.html
157. What is Batch Normalization In Deep Learning? | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/what-is-batch-normalization-in-deep-learning/
158. BatchNormalization layer - Keras, fecha de acceso: mayo 2, 2025, https://keras.io/api/layers/normalization_layers/batch_normalization/
159. Batch Norm Explained Visually - How it works, and why neural networks need it, fecha de acceso: mayo 2, 2025, https://towardsdatascience.com/batch-norm-explained-visually-how-it-works-and-why-neural-networks-need-it-b18919692739/
160. [1502.03167] Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift - arXiv, fecha de acceso: mayo 2, 2025, https://arxiv.org/abs/1502.03167
161. BatchNorm1d — PyTorch 2.7 documentation, fecha de acceso: mayo 2, 2025, https://pytorch.org/docs/stable/generated/torch.nn.BatchNorm1d.html
162. Batch Normalization in Deep Learning | Batch Learning in Keras - YouTube, fecha de acceso: mayo 2, 2025, https://m.youtube.com/watch?v=2AscwXePlnA&pp=ygUMI25vcm1pemF0aW9u
163. Introduction to Batch Normalization: Understanding the Basics - Analytics Vidhya, fecha de acceso: mayo 2, 2025, https://www.analyticsvidhya.com/blog/2021/03/introduction-to-batch-normalization/
164. Understanding the backward pass through Batch Normalization Layer, fecha de acceso: mayo 2, 2025, https://kratzert.github.io/2016/02/12/understanding-the-gradient-flow-through-the-batch-normalization-layer.html
165. Counterpropagation network with 2 inputs, 4 patterns and 3 outputs 1)... - ResearchGate, fecha de acceso: mayo 2, 2025, https://www.researchgate.net/figure/Counterpropagation-network-with-2-inputs-4-patterns-and-3-outputs-1-Set-the-weights-on_fig2_224244615
166. Weight Initialization Techniques in Neural Networks | Pinecone, fecha de acceso: mayo 2, 2025, https://www.pinecone.io/learn/weight-initialization/
167. Weight initialization - Wikipedia, fecha de acceso: mayo 2, 2025, https://en.wikipedia.org/wiki/Weight_initialization
168. Weight Initialization for Deep Learning Neural Networks - MachineLearningMastery.com, fecha de acceso: mayo 2, 2025, https://machinelearningmastery.com/weight-initialization-for-deep-learning-neural-networks/

169. How to Initialize Weights in Neural Networks? - Analytics Vidhya, fecha de acceso: mayo 2, 2025, https://www.analyticsvidhya.com/blog/2021/05/how-to-initialize-weights-in-neural-networks/
170. Initializing neural networks - DeepLearning.AI, fecha de acceso: mayo 2, 2025, https://www.deeplearning.ai/ai-notes/initialization/
171. Weight Initialization Techniques for Deep Neural Networks - GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/weight-initialization-techniques-for-deep-neural-networks/
172. A Gentle Introduction To Weight Initialization for Neural Networks ..., fecha de acceso: mayo 2, 2025, https://wandb.ai/sauravmaheshkar/initialization/reports/A-Gentle-Introduction-To-Weight-Initialization-for-Neural-Networks--Vmlldzo2ODExMTg
173. What are the techniques used to initialize weights for neural networks? - AI Stack Exchange, fecha de acceso: mayo 2, 2025, https://ai.stackexchange.com/questions/42975/what-are-the-techniques-used-to-initialize-weights-for-neural-networks
174. An Effective Weight Initialization Method for Deep Learning: Application to Satellite Image Classification - arXiv, fecha de acceso: mayo 2, 2025, https://arxiv.org/html/2406.00348v1
175. Xavier Initialization and Regularization - CS230 Deep Learning, fecha de acceso: mayo 2, 2025, https://cs230.stanford.edu/section/4/
176. Understanding weight initialization for neural networks - PyImageSearch, fecha de acceso: mayo 2, 2025, https://pyimagesearch.com/2021/05/06/understanding-weight-initialization-for-neural-networks/
177. Xavier Initialization Explained - Papers With Code, fecha de acceso: mayo 2, 2025, https://paperswithcode.com/method/xavier-initialization
178. Xavier initialization | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/xavier-initialization/
179. What is the numpy equivalent of TensorFlow Xavier initializer for CNN? - Stack Overflow, fecha de acceso: mayo 2, 2025, https://stackoverflow.com/questions/62249084/what-is-the-numpy-equivalent-of-tensorflow-xavier-initializer-for-cnn
180. Default Weight Initialization vs Xavier Initialization - PyTorch Forums, fecha de acceso: mayo 2, 2025, https://discuss.pytorch.org/t/default-weight-initialization-vs-xavier-initialization/50691
181. torch.nn.init — PyTorch 2.7 documentation, fecha de acceso: mayo 2, 2025, https://pytorch.org/docs/stable/nn.init.html
182. Xavier Glorot Initialization in Neural Networks - Math Proof - Towards Data Science, fecha de acceso: mayo 2, 2025, https://towardsdatascience.com/xavier-glorot-initialization-in-neural-networks-math-proof-4682bf5c6ec3/

183. Weight Initialization Schemes - Xavier (Glorot) and He | Mustafa Murat ARAT, fecha de acceso: mayo 2, 2025, https://mmuratarat.github.io/2019-02-25/xavier-glorot-he-weight-init
184. Glorot initialization, the tricks that led to the current success of Deep Learning. - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/learnmachinelearning/comments/1435lly/glorot_initialization_the_tricks_that_led_to_the/
185. What Is Xavier Initialization? - 365 Data Science, fecha de acceso: mayo 2, 2025, https://365datascience.com/tutorials/machine-learning-tutorials/what-is-xavier-initialization/
186. Where Does the Normal Glorot Initialization Come from? - Data Science Stack Exchange, fecha de acceso: mayo 2, 2025, https://datascience.stackexchange.com/questions/102036/where-does-the-normal-glorot-initialization-come-from
187. What is glorot uniform? - keras - Data Science Stack Exchange, fecha de acceso: mayo 2, 2025, https://datascience.stackexchange.com/questions/102733/what-is-glorot-uniform
188. Kaiming Initialization Explained - Papers With Code, fecha de acceso: mayo 2, 2025, https://paperswithcode.com/method/he-initialization
189. Kaiming Initialization in Deep Learning - GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/kaiming-initialization-in-deep-learning/
190. Layer weight initializers - Keras, fecha de acceso: mayo 2, 2025, https://keras.io/api/layers/initializers/
191. Why does He initialization work ? : r/deeplearning - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/deeplearning/comments/qn8cgb/why_does_he_initialization_work/
192. What is initialization='he' equivalent in PyTorch, fecha de acceso: mayo 2, 2025, https://discuss.pytorch.org/t/what-is-initialization-he-equivalent-in-pytorch/117388
193. Neural Networks throw their weights around | Xavier & He Initialization | Deep Learning basics - YouTube, fecha de acceso: mayo 2, 2025, https://www.youtube.com/watch?v=LKWatKGRZLI
194. Optimization Algorithms in Neural Networks - KDnuggets, fecha de acceso: mayo 2, 2025, https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html
195. Deep Learning Optimization Algorithms - Neptune.ai, fecha de acceso: mayo 2, 2025, https://neptune.ai/blog/deep-learning-optimization-algorithms
196. Optimization Rule in Deep Neural Networks - GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/optimization-rule-in-deep-neural-networks/
197. Optimizers in Deep Learning: A Detailed Guide - Analytics Vidhya, fecha de

acceso: mayo 2, 2025,
https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/
198. 5 algorithms to train a neural network, fecha de acceso: mayo 2, 2025,
https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network/
199. Optimization algorithms underlying neural networks - Lund University Publications, fecha de acceso: mayo 2, 2025,
https://lup.lub.lu.se/student-papers/record/9112234/file/9116796.pdf
200. 12. Optimization Algorithms - Dive into Deep Learning, fecha de acceso: mayo 2, 2025, http://www.d2l.ai/chapter_optimization/
201. Neural Network Optimizers from Scratch in Python - Towards Data Science, fecha de acceso: mayo 2, 2025,
https://towardsdatascience.com/neural-network-optimizers-from-scratch-in-python-af76ee087aab/
202. Complete Guide to the Adam Optimization Algorithm | Built In, fecha de acceso: mayo 2, 2025, https://builtin.com/machine-learning/adam-optimization
203. What is Adam Optimizer? | GeeksforGeeks, fecha de acceso: mayo 2, 2025,
https://www.geeksforgeeks.org/adam-optimizer/
204. Why does Adam optimizer work so well? : r/learnmachinelearning - Reddit, fecha de acceso: mayo 2, 2025,
https://www.reddit.com/r/learnmachinelearning/comments/1gbqci5/why_does_adam_optimizer_work_so_well/
205. SGD with Momentum Explained | Papers With Code, fecha de acceso: mayo 2, 2025, https://paperswithcode.com/method/sgd-with-momentum
206. Papers with Code - SGD with Momentum Explained, fecha de acceso: mayo 2, 2025,
https://www.khoury.northeastern.edu/home/vip/teach/DMcourse/3_dim_reduction/tSNE/tsne_matlab/SGD_Momentum.pdf
207. Stochastic gradient descent - Wikipedia, fecha de acceso: mayo 2, 2025,
https://en.wikipedia.org/wiki/Stochastic_gradient_descent
208. SGD - Keras, fecha de acceso: mayo 2, 2025,
https://keras.io/api/optimizers/sgd/
209. Gradient Descent With Momentum from Scratch - MachineLearningMastery.com, fecha de acceso: mayo 2, 2025,
https://machinelearningmastery.com/gradient-descent-with-momentum-from-scratch/
210. An Improved Analysis of Stochastic Gradient Descent with Momentum, fecha de acceso: mayo 2, 2025,
https://proceedings.neurips.cc/paper/2020/file/d3f5d4de09ea19461dab00590df91e4f-Paper.pdf
211. [2007.07989] An Improved Analysis of Stochastic Gradient Descent with Momentum - arXiv, fecha de acceso: mayo 2, 2025,
https://arxiv.org/abs/2007.07989
212. SGD — PyTorch 2.7 documentation, fecha de acceso: mayo 2, 2025,
https://pytorch.org/docs/stable/generated/torch.optim.SGD.html

213. SGD with momentum - why the formula change? - PyTorch Forums, fecha de acceso: mayo 2, 2025, https://discuss.pytorch.org/t/sgd-with-momentum-why-the-formula-change/135444
214. pytorch optim.SGD with momentum how to check "velocity"? - Stack Overflow, fecha de acceso: mayo 2, 2025, https://stackoverflow.com/questions/69465707/pytorch-optim-sgd-with-momentum-how-to-check-velocity
215. Adam Optimizer Tutorial: Intuition and Implementation in Python | DataCamp, fecha de acceso: mayo 2, 2025, https://www.datacamp.com/tutorial/adam-optimizer-tutorial
216. Understanding the RMSprop Optimizer: A Guide - Built In, fecha de acceso: mayo 2, 2025, https://builtin.com/articles/rmsprop-optimizer
217. RMSprop Optimizer Tutorial: Intuition and Implementation in Python - DataCamp, fecha de acceso: mayo 2, 2025, https://www.datacamp.com/tutorial/rmsprop-optimizer-tutorial
218. What is RMSProp? Principles & Advantages - Deepchecks, fecha de acceso: mayo 2, 2025, https://www.deepchecks.com/glossary/rmsprop/
219. Gradient Descent With RMSProp from Scratch | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/gradient-descent-with-rmsprop-from-scratch/
220. RMSProp Optimizer in Deep Learning | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/rmsprop-optimizer-in-deep-learning/
221. RMSprop - Keras, fecha de acceso: mayo 2, 2025, https://keras.io/api/optimizers/rmsprop/
222. Understanding Deep Learning Optimizers: Momentum, AdaGrad, RMSProp & Adam, fecha de acceso: mayo 2, 2025, https://towardsdatascience.com/understanding-deep-learning-optimizers-momentum-adagrad-rmsprop-adam-e311e377e9c2/
223. RMSProp - Cornell University Computational Optimization Open Textbook, fecha de acceso: mayo 2, 2025, https://optimization.cbe.cornell.edu/index.php?title=RMSProp
224. tf.keras.optimizers.RMSprop | TensorFlow v2.16.1, fecha de acceso: mayo 2, 2025, https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/RMSprop
225. RMSprop — PyTorch 2.7 documentation, fecha de acceso: mayo 2, 2025, https://pytorch.org/docs/stable/generated/torch.optim.RMSprop.html
226. RMSProp Optimizer in Deep Learning | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/rmsprop-optimizer-in-deep-learning/?ref=asr2
227. RMSprop Optimizer Explained in Detail | Deep Learning - YouTube, fecha de acceso: mayo 2, 2025, https://m.youtube.com/watch?v=ajI_HTyaCu8
228. Why are optimization algorithms for deep learning so simple? - AI Stack Exchange, fecha de acceso: mayo 2, 2025, https://ai.stackexchange.com/questions/32140/why-are-optimization-algorithms-

for-deep-learning-so-simple
229. Adam - Keras, fecha de acceso: mayo 2, 2025, https://keras.io/api/optimizers/adam/
230. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning - MachineLearningMastery.com, fecha de acceso: mayo 2, 2025, https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/
231. Adam — PyTorch 2.7 documentation, fecha de acceso: mayo 2, 2025, https://pytorch.org/docs/stable/generated/torch.optim.Adam.html
232. [1412.6980] Adam: A Method for Stochastic Optimization - arXiv, fecha de acceso: mayo 2, 2025, https://arxiv.org/abs/1412.6980
233. Who's Adam and What's He Optimizing? | Deep Dive into Optimizers for Machine Learning!, fecha de acceso: mayo 2, 2025, https://www.youtube.com/watch?v=MD2fYip6QsQ
234. tf.keras.optimizers.Adam | TensorFlow v2.16.1, fecha de acceso: mayo 2, 2025, https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam
235. Analysis of the modular topology of hybrid neural networks - CEUR-WS.org, fecha de acceso: mayo 2, 2025, https://ceur-ws.org/Vol-2683/paper5.pdf
236. Adam Optimizer for Neural Networks | Faster, Smarter Deep Learning - YouTube, fecha de acceso: mayo 2, 2025, https://www.youtube.com/watch?v=Z-1bH6bGcLw
237. A Robust Deep Learning Model for Financial Distress Prediction - The Science and Information (SAI) Organization, fecha de acceso: mayo 2, 2025, https://thesai.org/Downloads/Volume11No2/Paper_22-A_Robust_Deep_Learning_Model.pdf
238. How to Choose a Learning Rate Scheduler for Neural Networks - neptune.ai, fecha de acceso: mayo 2, 2025, https://neptune.ai/blog/how-to-choose-a-learning-rate-scheduler
239. Learning Rate in Neural Network - GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/impact-of-learning-rate-on-a-model/
240. Using Learning Rate Schedule in PyTorch Training - MachineLearningMastery.com, fecha de acceso: mayo 2, 2025, https://machinelearningmastery.com/using-learning-rate-schedule-in-pytorch-training/
241. 12.11. Learning Rate Scheduling — Dive into Deep Learning 1.0.3 documentation, fecha de acceso: mayo 2, 2025, https://d2l.ai/chapter_optimization/lr-scheduler.html
242. what is learning rate in neural network? - Data Science Stack Exchange, fecha de acceso: mayo 2, 2025, https://datascience.stackexchange.com/questions/69917/what-is-learning-rate-in-neural-network
243. Is it good learning rate for Adam method? - Stack Overflow, fecha de acceso: mayo 2, 2025, https://stackoverflow.com/questions/42966393/is-it-good-learning-rate-for-adam-method

244. [D] Are there any comparison studies on learning rate schedules for generative transformers? : r/MachineLearning - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/MachineLearning/comments/u1nt7m/d_are_there_any_comparison_studies_on_learning/
245. How to set Learning Rate for a Neural Network? - PyTorch Forums, fecha de acceso: mayo 2, 2025, https://discuss.pytorch.org/t/how-to-set-learning-rate-for-a-neural-network/43362
246. [D] How to pick a learning rate scheduler? : r/MachineLearning - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/MachineLearning/comments/oy3co1/d_how_to_pick_a_learning_rate_scheduler/
247. Learning Rate Decay | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/learning-rate-decay/
248. Learning Rate Scheduling - Deep Learning Wizard, fecha de acceso: mayo 2, 2025, https://www.deeplearningwizard.com/deep_learning/boosting_models_pytorch/lr_scheduling/
249. Learning Rate Schedules - Kaggle, fecha de acceso: mayo 2, 2025, https://www.kaggle.com/code/ritvik1909/learning-rate-schedules
250. Cosine Annealing In Machine Learning Simplified: Understand How It Works, fecha de acceso: mayo 2, 2025, https://spotintelligence.com/2024/04/29/cosine-annealing-in-machine-learning/
251. What are the advantages and disadvantages of using a cosine annealing learning rate schedule? - Massed Compute, fecha de acceso: mayo 2, 2025, https://massedcompute.com/faq-answers/?question=What+are+the+advantages+and+disadvantages+of+using+a+cosine+annealing+learning+rate+schedule%3F
252. Learning Rate Scheduler | Continuum Labs, fecha de acceso: mayo 2, 2025, https://training.continuumlabs.ai/training/the-fine-tuning-process/hyperparameters/learning-rate-scheduler
253. What is the Plateau Problem in Neural Networks and How to Fix it? | AIM Media House, fecha de acceso: mayo 2, 2025, https://analyticsindiamag.com/ai-trends/what-is-the-plateau-problem-in-neural-networks-and-how-to-fix-it/
254. Step Decay Schedules - ApX Machine Learning, fecha de acceso: mayo 2, 2025, https://apxml.com/courses/deep-learning-regularization-optimization/chapter-7-optimization-refinements-tuning/step-decay-lr-schedule
255. Step Decay Explained - Papers With Code, fecha de acceso: mayo 2, 2025, https://paperswithcode.com/method/step-decay
256. Keras learning rate schedules and decay - PyImageSearch, fecha de acceso: mayo 2, 2025, https://pyimagesearch.com/2019/07/22/keras-learning-rate-schedules-and-decay/

257. Day 26: Learning Rate Schedules - DEV Community, fecha de acceso: mayo 2, 2025, https://dev.to/nareshnishad/day-26-learning-rate-schedules-3922
258. The Step Decay Schedule: A Near Optimal, Geometrically Decaying Learning Rate Procedure For Least Squares, fecha de acceso: mayo 2, 2025, https://proceedings.neurips.cc/paper/2019/hash/2f4059ce1227f021edc5d9c6f0f17dc1-Abstract.html
259. How to do exponential learning rate decay in PyTorch? - autograd, fecha de acceso: mayo 2, 2025, https://discuss.pytorch.org/t/how-to-do-exponential-learning-rate-decay-in-pytorch/63146
260. StepLR — PyTorch 2.7 documentation, fecha de acceso: mayo 2, 2025, https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.StepLR.html
261. Decreasing the learning rate - PyTorch Forums, fecha de acceso: mayo 2, 2025, https://discuss.pytorch.org/t/decreasing-the-learning-rate/155802
262. Weight Decay on custom learning rate scheduler (with Pytorch) : r/learnmachinelearning, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/learnmachinelearning/comments/x0j33q/weight_decay_on_custom_learning_rate_scheduler/
263. Pytorch Change the learning rate based on number of epochs - Stack Overflow, fecha de acceso: mayo 2, 2025, https://stackoverflow.com/questions/60050586/pytorch-change-the-learning-rate-based-on-number-of-epochs
264. Exponential decay learning rate - vision - PyTorch Forums, fecha de acceso: mayo 2, 2025, https://discuss.pytorch.org/t/exponential-decay-learning-rate/76384
265. Implement learning rate decay - PyTorch Forums, fecha de acceso: mayo 2, 2025, https://discuss.pytorch.org/t/implement-learning-rate-decay/98238
266. Keras learning rate decay in pytorch - python - Stack Overflow, fecha de acceso: mayo 2, 2025, https://stackoverflow.com/questions/55663375/keras-learning-rate-decay-in-pytorch
267. ExponentialDecay - Keras, fecha de acceso: mayo 2, 2025, https://keras.io/api/optimizers/learning_rate_schedules/exponential_decay/
268. ExponentialDecay - Keras, fecha de acceso: mayo 2, 2025, https://keras.io/2/api/optimizers/learning_rate_schedules/exponential_decay/
269. Exponential Decay Explained - Papers With Code, fecha de acceso: mayo 2, 2025, https://paperswithcode.com/method/exponential-decay
270. A LearningRateSchedule that uses an exponential decay schedule. — learning_rate_schedule_exponential_decay - R Interface to Keras, fecha de acceso: mayo 2, 2025, https://keras3.posit.co/reference/learning_rate_schedule_exponential_decay.html
271. [1910.07454] An Exponential Learning Rate Schedule for Deep Learning - arXiv, fecha de acceso: mayo 2, 2025, https://arxiv.org/abs/1910.07454
272. [D] How does one choose a learning rate schedule for models that take days or weeks to train? - Reddit, fecha de acceso: mayo 2, 2025,

https://www.reddit.com/r/MachineLearning/comments/xeyzf7/d_how_does_one_choose_a_learning_rate_schedule/
273. Cosine Annealing Explained | Papers With Code, fecha de acceso: mayo 2, 2025, https://paperswithcode.com/method/cosine-annealing
274. CosineAnnealingLR - Computer Vision Wiki - CloudFactory, fecha de acceso: mayo 2, 2025, https://wiki.cloudfactory.com/docs/mp-wiki/scheduler/cosineannealinglr
275. CosineAnnealingLR — mmengine 0.10.7 documentation, fecha de acceso: mayo 2, 2025, https://mmengine.readthedocs.io/en/stable/api/generated/mmengine.optim.CosineAnnealingLR.html
276. CosineAnnealingLR — PyTorch 2.7 documentation, fecha de acceso: mayo 2, 2025, https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.CosineAnnealingLR.html
277. CosineAnnealingWarmRestarts — PyTorch 2.7 documentation, fecha de acceso: mayo 2, 2025, https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.CosineAnnealingWarmRestarts.html
278. Unit 6.2 - Learning Rates and Learning Rate Schedulers - Lightning AI, fecha de acceso: mayo 2, 2025, https://lightning.ai/courses/deep-learning-fundamentals/unit-6-overview-essential-deep-learning-tips-tricks/unit-6.2-learning-rates-and-learning-rate-schedulers/
279. How to Properly Use PyTorch's CosineAnnealingWarmRestarts Scheduler | tips - Wandb, fecha de acceso: mayo 2, 2025, https://wandb.ai/wandb_fc/tips/reports/How-to-Properly-Use-PyTorch-s-CosineAnnealingWarmRestarts-Scheduler--VmlldzoyMTA3MjM2
280. Using both learning rate warm up and a learning rate scheduler - PyTorch Forums, fecha de acceso: mayo 2, 2025, https://discuss.pytorch.org/t/using-both-learning-rate-warm-up-and-a-learning-rate-scheduler/177767
281. CosineAnnealingLR step size (T_max) - PyTorch Forums, fecha de acceso: mayo 2, 2025, https://discuss.pytorch.org/t/cosineannealinglr-step-size-t-max/104687
282. How to use Cosine Annealing? - PyTorch Forums, fecha de acceso: mayo 2, 2025, https://discuss.pytorch.org/t/how-to-use-cosine-annealing/91522
283. Using both Learning Rate Warmup and Learning Rate Decay Schedule in PyTorch - Reddit, fecha de acceso: mayo 2, 2025, https://www.reddit.com/r/deeplearning/comments/12pkimm/using_both_learning_rate_warmup_and_learning_rate/
284. Cosine Learning Rate Decay - vision - PyTorch Forums, fecha de acceso: mayo 2, 2025, https://discuss.pytorch.org/t/cosine-learning-rate-decay/145276
285. Guide to Pytorch Learning Rate Scheduling | Kaggle, fecha de acceso: mayo 2, 2025,

https://www.kaggle.com/code/isbhargav/guide-to-pytorch-learning-rate-scheduling
286. www.ibm.com, fecha de acceso: mayo 2, 2025, https://www.ibm.com/think/topics/ensemble-learning#:~:text=Ensemble%20learning%20is%20a%20machine,than%20a%20single%20model%20alone.
287. What is ensemble learning? - IBM, fecha de acceso: mayo 2, 2025, https://www.ibm.com/think/topics/ensemble-learning
288. Ensemble Models: What Are They and When Should You Use Them? - Built In, fecha de acceso: mayo 2, 2025, https://builtin.com/machine-learning/ensemble-model
289. A Comprehensive Guide to Ensemble Learning (with Python codes) - Analytics Vidhya, fecha de acceso: mayo 2, 2025, https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/
290. Ensemble learning - Wikipedia, fecha de acceso: mayo 2, 2025, https://en.wikipedia.org/wiki/Ensemble_learning
291. A Comprehensive Guide to Ensemble Learning: What Exactly Do You Need to Know, fecha de acceso: mayo 2, 2025, https://neptune.ai/blog/ensemble-learning-guide
292. 3 Primary Ensemble Methods to Enhance an ML Model's Accuracy - Data Science Dojo, fecha de acceso: mayo 2, 2025, https://datasciencedojo.com/blog/ensemble-methods-in-machine-learning/
293. Ensemble Learning | GeeksforGeeks, fecha de acceso: mayo 2, 2025, https://www.geeksforgeeks.org/a-comprehensive-guide-to-ensemble-learning/
294. Ensemble Learning Techniques Tutorial - Kaggle, fecha de acceso: mayo 2, 2025, https://www.kaggle.com/code/pavansanagapati/ensemble-learning-techniques-tutorial
295. 1.11. Ensembles: Gradient boosting, random forests, bagging, voting, stacking - Scikit-learn, fecha de acceso: mayo 2, 2025, https://scikit-learn.org/stable/modules/ensemble.html
296. A Gentle Introduction to Ensemble Learning Algorithms - MachineLearningMastery.com, fecha de acceso: mayo 2, 2025, https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/