Efficient Inference in Deep Learning

Sebastien Perre

University of Victoria
CSC 499: Honours Seminar and Project
Professor George Tzanetakis

July 22, 2025

Outline

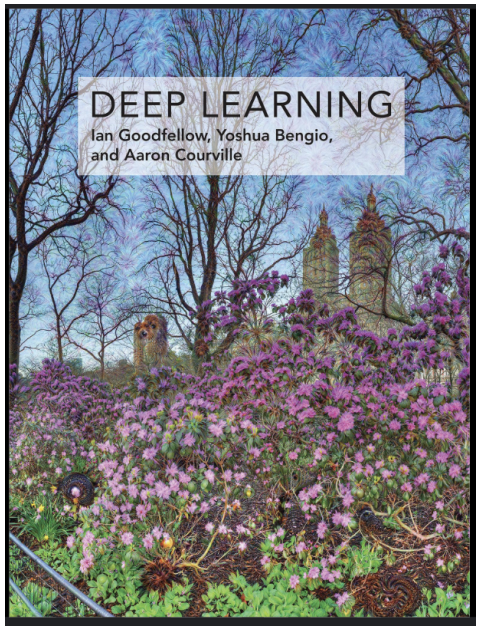
LEARNING OVERVIEW

Learning Overview: Introduction/Motivation

I have touched PyTorch only a few times, so to have a robust understanding of what I am doing, I have to learn how to use PyTorch and deep learning concepts.

Learning Overview: Introduction

(a) PyTorch Crash Course



(b) Deep Learning Book

Learning Overview: Deep Learning Framework

Figure: PyTorch Logo

Learning Overview: Deep Learning Framework

PyTorch Motivation:

1. Ease of Use
2. Quick Creation of Deep Learning Models
3. Strong NumPy Integration and Community Support
4. GPU Acceleration (Nvidia GPU Support)

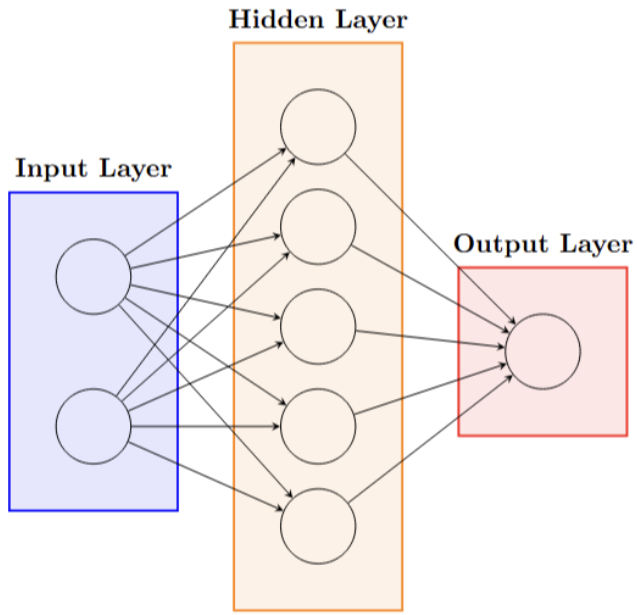
Learning Overview: Deep Learning Framework

```python

class Model(nn.Module):
    def __init__(self):
        super().__init__()
        self.layer_1 = nn.Linear(in_features = 2, out_features = 5)
        self.layer_2 = nn.Linear(in_features = 5, out_features = 1)

    def forward(self, x):
        return self.layer_2(self.layer_1(x))

```



Learning Overview: YouTube Course

Chapter Overview:

1. Chapter 0: Fundamentals
2. Chapter 1: Workflow
3. Chapter 2: Neural Network Classification
4. Chapter 3: Computer Vision
5. Chapter 4: Custom Datasets

Learning Overview: YouTube Course

Chapter 0: Fundamentals

- Introduction & Motivation for Deep Learning and PyTorch
- Introduction to Tensors
- Creating Tensors
- Getting information from tensors
- Manipulating tensors
- Dealing with tensor shapes
- Indexing on tensors
- Mixing PyTorch tensors and NumPy
- Reproducibility
- Running tensors on GPU

Learning Overview: YouTube Course

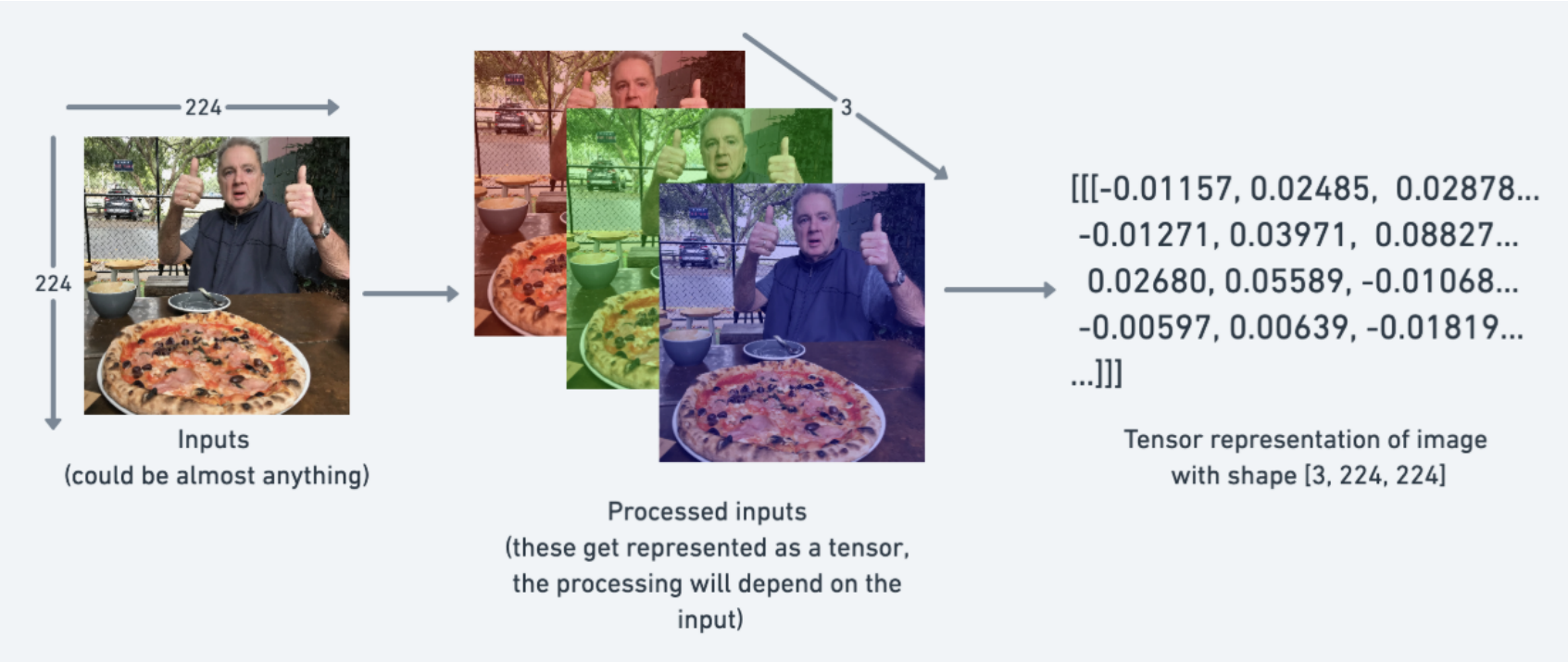


Figure: Example of an image represented as a tensor.

.

Learning Overview: YouTube Course

Chapter 1: Workflow

- Getting Data Ready
- Building a Model
- Fitting the Model to Data
- Making Predictions and Evaluating the Model
- Saving and Loading a Model

Learning Overview: YouTube Course

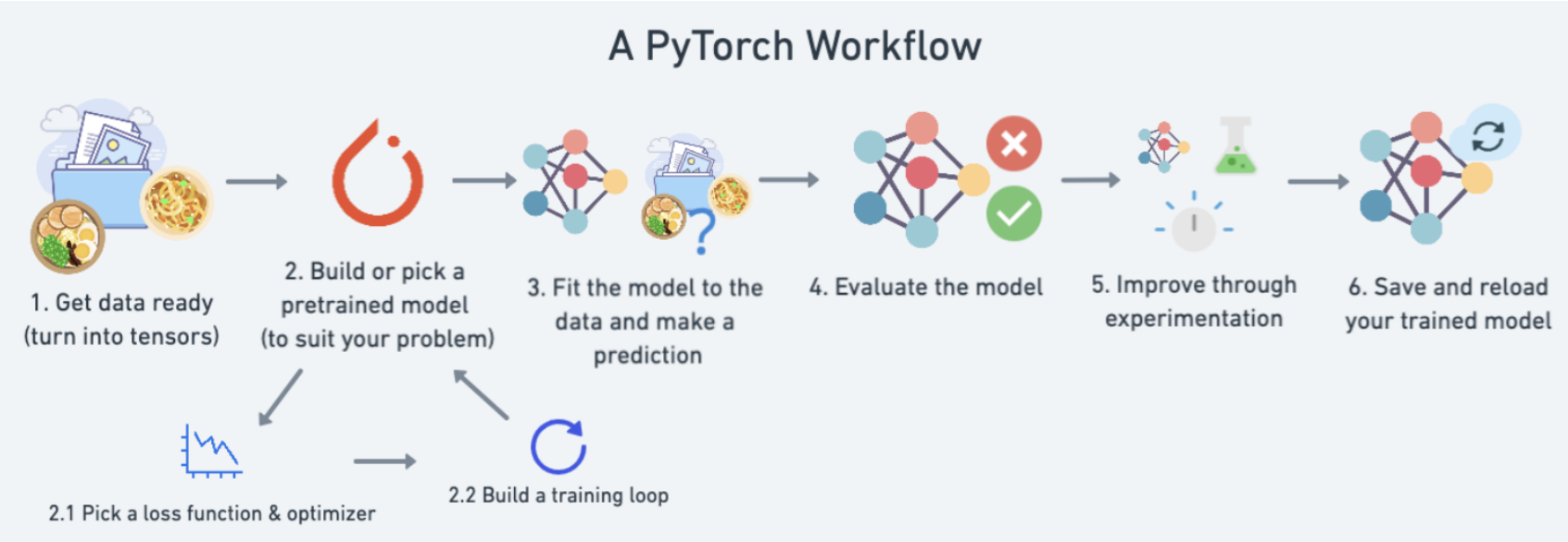


Figure: A PyTorch Workflow.

Learning Overview: YouTube Course

Chapter 2: Neural Network Classification

- Binary Classification
- Multi-Class Classification
- Multi-Label Classification

Learning Overview: YouTube Course

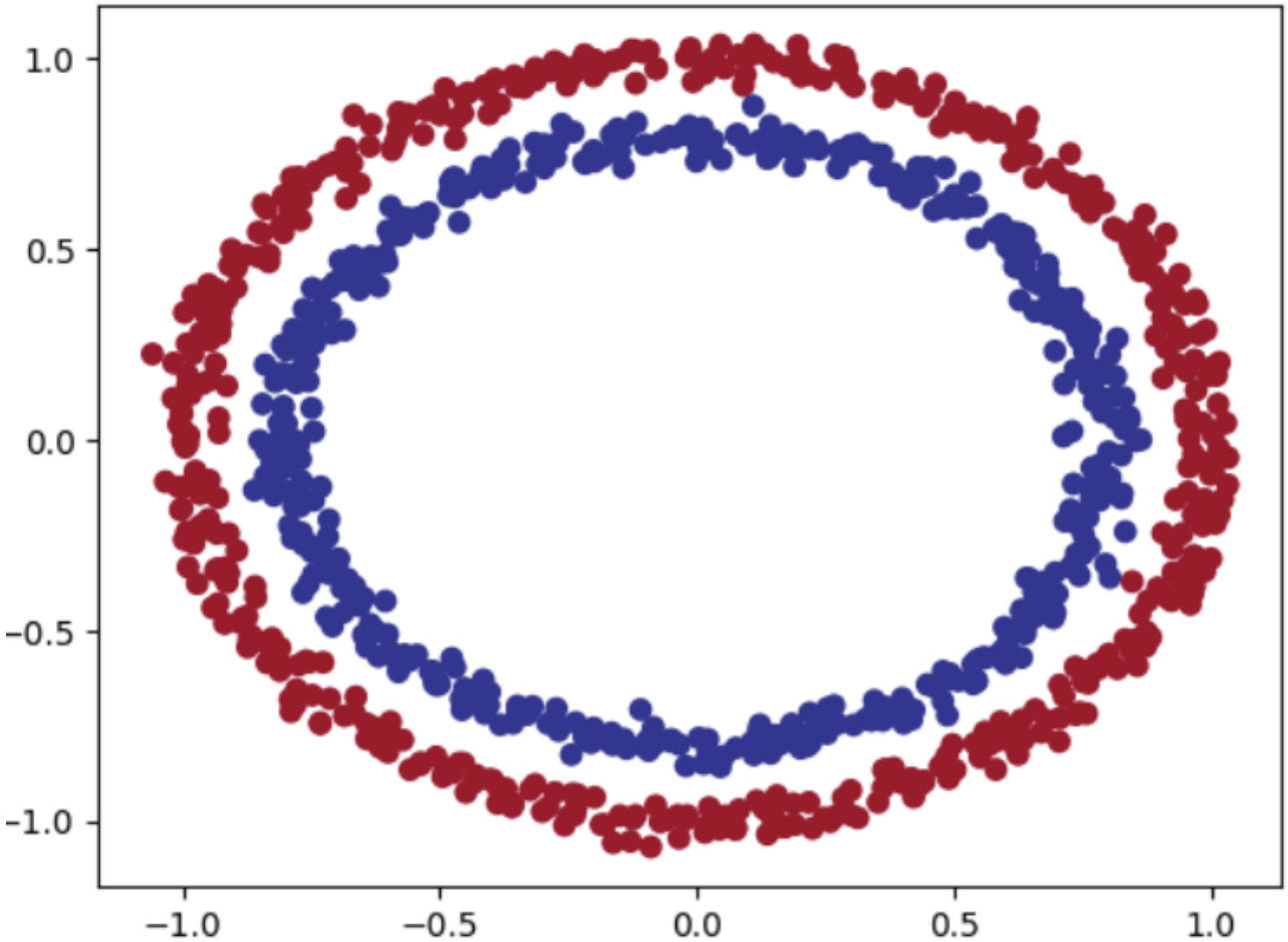


Figure: The Classification Problem I Solve in Ch. 2.

Learning Overview: YouTube Course

Chapter 3: Computer Vision

- Computer Vision Libraries in PyTorch
- Loading Image Data
- Preparing Image Data
- Building a Baseline Model
- Making Predictions and Evaluating a Model
- Setting up Device Agnostic Code
- Convolutional Neural Networks (CNN)
- Comparing Models
- Evaluating Models
- Confusion Matrices

Learning Overview: YouTube Course

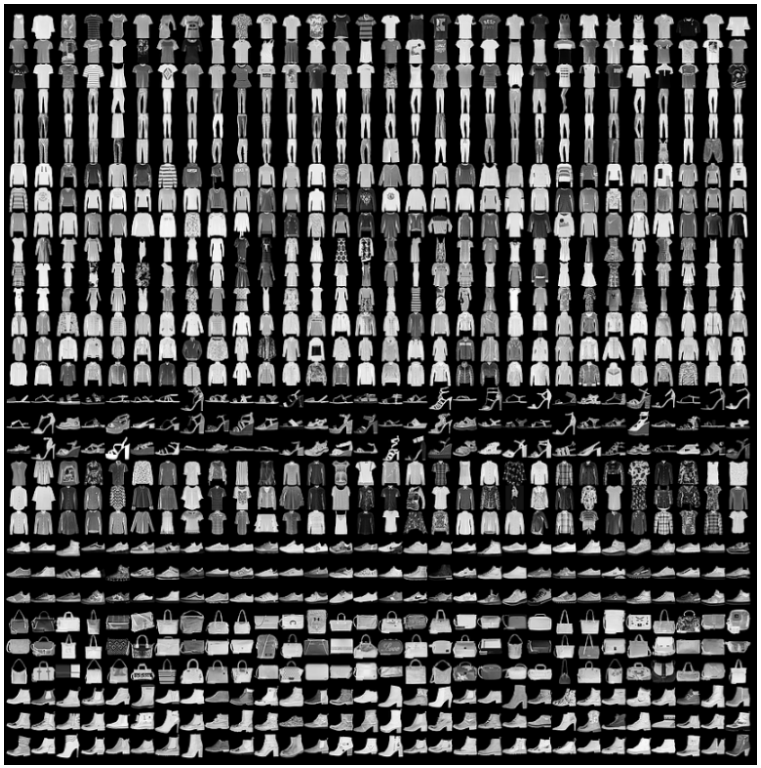


Figure: Samples from the Fashion MNIST dataset

.

Learning Overview: YouTube Course

Chapter 4: Custom Datasets

- PyTorch Custom Data Loading Libraries
- Data Augmentation
- Tiny VGGs
- Loss Curves

Learning Overview: Deep Learning Book

Chapter Overview

1. Chapter 1: Introduction
2. Chapter 2: Linear Algebra
3. Chapter 3: Probability and Information Theory
4. Chapter 4: Numerical Conditioning
5. Chapter 5: Machine Learning Basics
6. Chapter 6: Deep Feedforward Networks
7. Chapter 7: Regularization
8. Chapter 8: Optimization
9. Chapter 9: Convolutional Networks
10. Chapter 10: Recursive and Recurrent Nets
11. Chapter 11: Practical Methodology
12. Chapter 12: Applications

Learning Overview: Deep Learning Book

Chapter 1: Introduction

- Motivation for Deep Learning
- Place in AI
- History

Learning Overview: Deep Learning

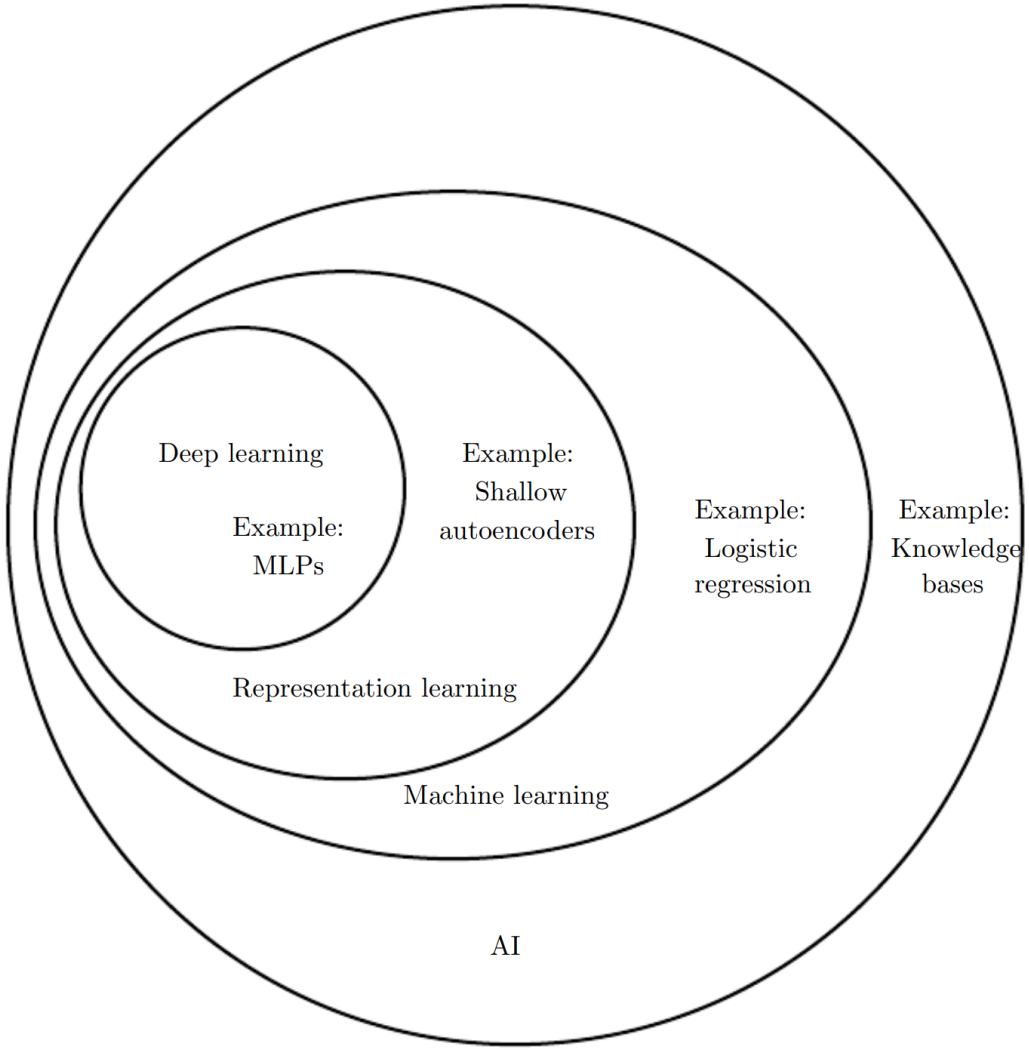


Figure: AI Venn Diagram. Taken from [1]

Learning Overview: Deep Learning Book

Chapter 2: Linear Algebra

- Scalars, Vectors, Matrices and Tensors
- Vector and Matrix Multiplication
- Identity and Inverse Matrices
- Linear Dependence, Span and Norms
- Symmetric and Orthogonal Matrices
- Eigendecomposition and Singular Value Decomposition (SVD)
- Moore-Penrose Pseudoinverse
- Trace and Determinants
- Principal Component Analysis (PCA)

Learning Overview: Deep Learning Book

Chapter 3: Probability and Information Theory

- Random Variables and Probability Distributions
- Marginal and Conditional Probability
- Chain Rule
- Independence and Conditional Independence
- Expectation, Variance and Covariance
- Bayes Rule
- Continuous Variables and Information Theory
- Entropy and KL Divergence
- Probabilistic Models

Learning Overview: Deep Learning Book

Chapter 4: Numerical Conditioning

- Underflow and Overflow
- Poor Conditioning
- Gradient-Based Optimization
- Constrained Optimization
- Linear Least Squares

Learning Overview: Deep Learning Book

Chapter 5: Machine Learning Basics

- Learning Algorithms
- Capacity, Overfitting, Underfitting, Hyperparameters, and Validation Sets
- Estimators, Bias, Variance, Maximum Likelihood Estimation
- Bayesian Statistics
- Stochastic Gradient Descent
- Challenges in Machine Learning

Learning Overview: Deep Learning Book

Chapter 6: Deep Feedforward Networks

- Feed-Forward Network
- Gradient Based Learning
- Architecture Design
- Backpropagation
- Historical Notes

Learning Overview: Deep Learning Book

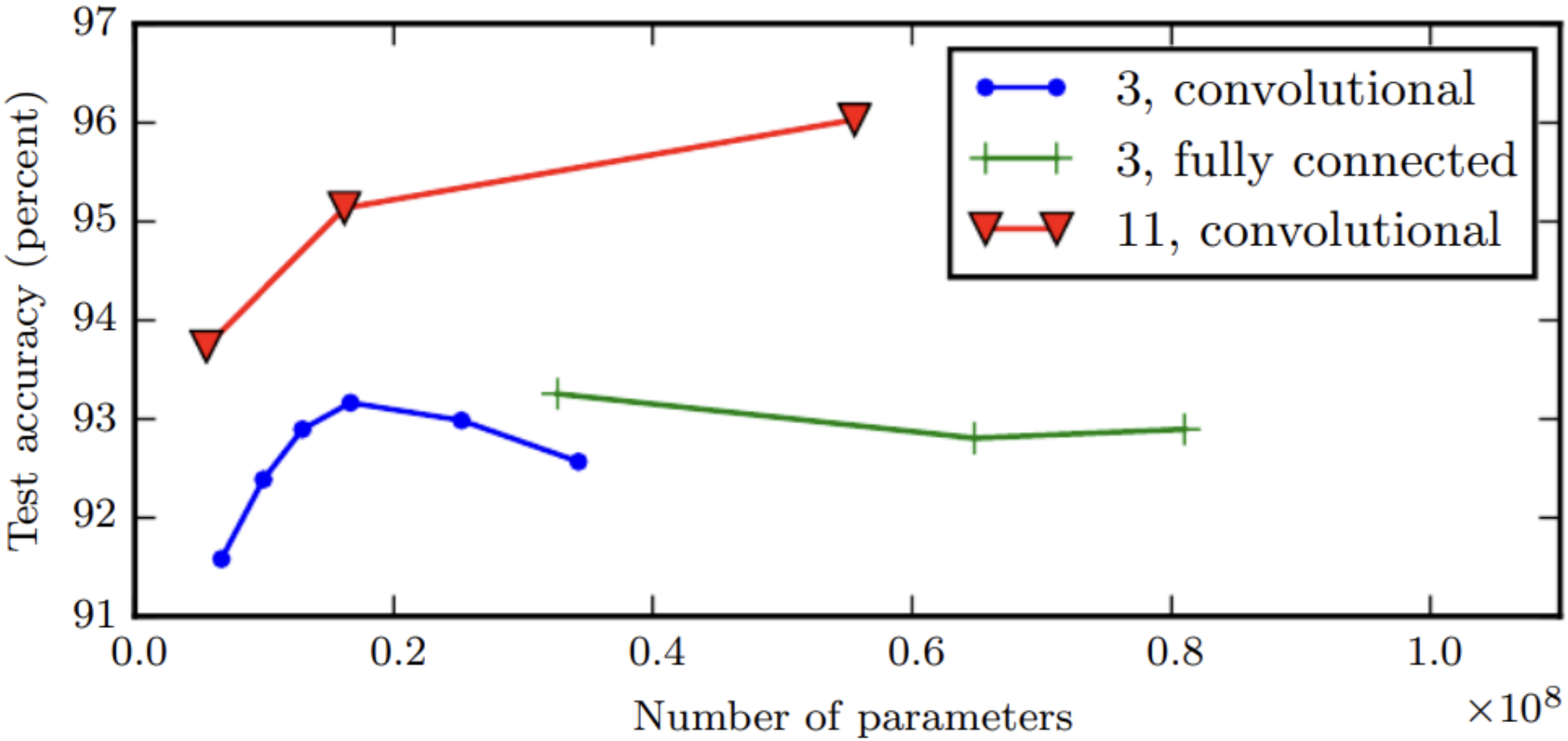


Figure: Graph illustrating the benefit of having deeper models.

Learning Overview: Deep Learning Book

Chapter 7: Regularization

- Parameter Norm Penalties
- Under-Constrained Problems
- Dataset Augmentation
- Semi-Supervised Learning and Multi-Task Learning
- Early Stopping
- Sparse Representations
- Ensemble Methods and Dropout
- Adversarial Training

Learning Overview: Deep Learning

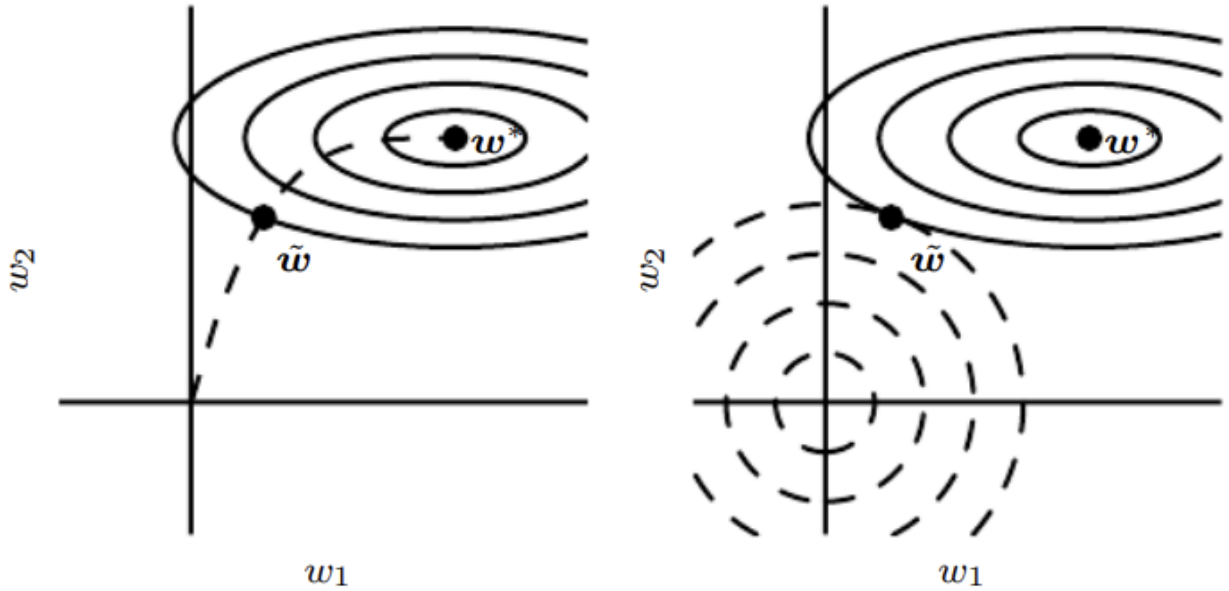


Figure: Graph illustrating early stopping. You see the model stops at a contour and not at the optimal solution w^* to avoid overfitting the data

.

Learning Overview: Deep Learning Book

Chapter 8: Optimization

- Vanishing and Exploding Gradients
- Parameter Initialization Strategies
- Momentum
- AdaGrad, RMSProp, and Adam
- Meta Algorithms

Learning Overview: Deep Learning Book

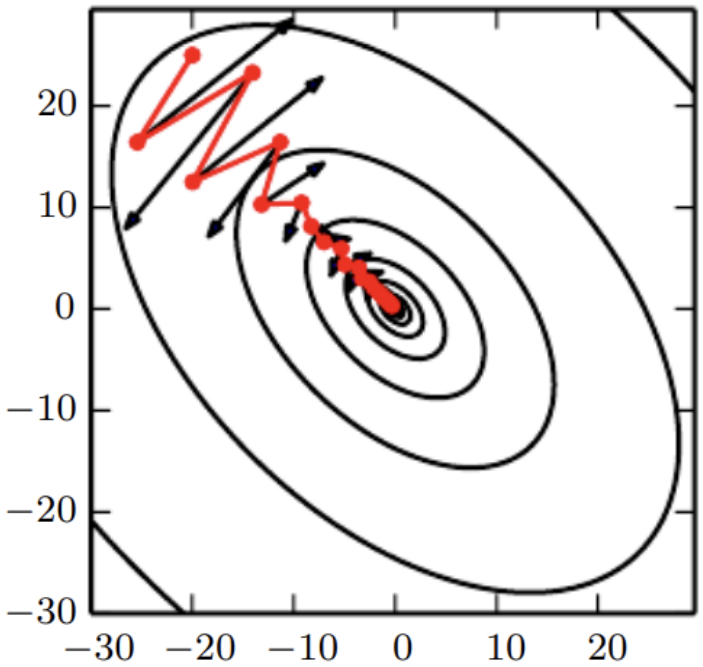


Figure: Graph illustrating momentum. The red line indicates the time steps using momentum while the black arrows represent if the model took steps from the original stochastic gradient descent.

Learning Overview: Deep Learning Book

Chapter 9: Convolutional Networks

- Convolutional Neural Networks (CNNs)
- Pooling
- Variants of Basic Convolution Function
- Advanced Topics
- Inspiration for Convolutional Neural Networks

Learning Overview: Deep Learning Book

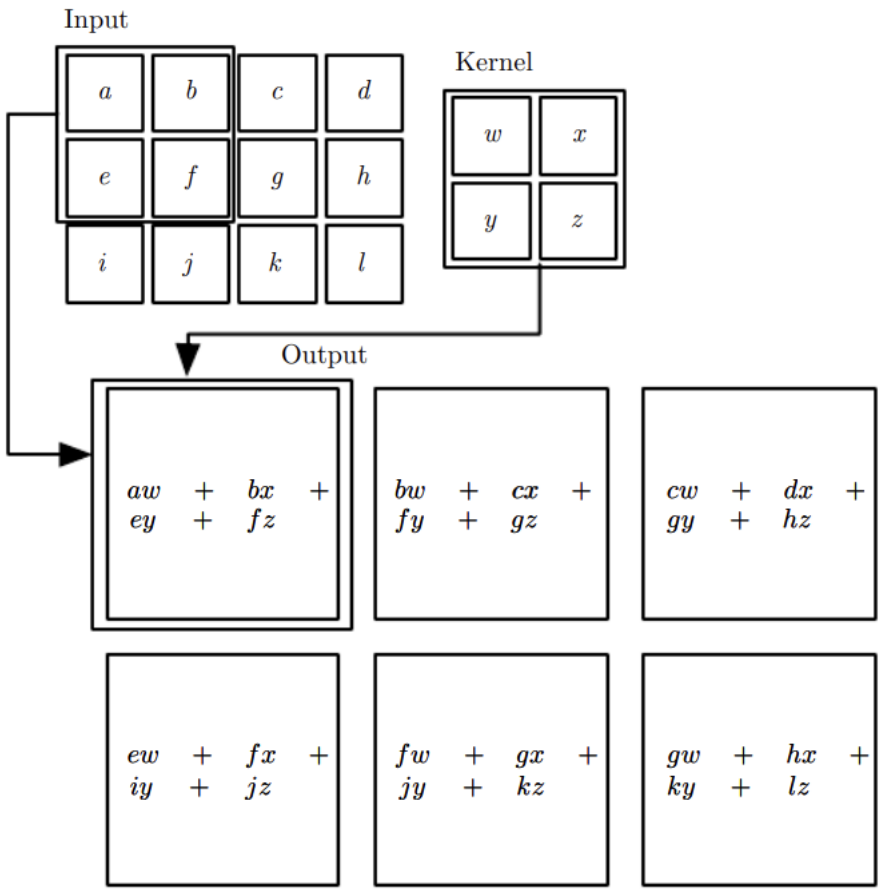


Figure: Example of a convolution applied on a 2D tensor. Taken from [1]

.

Learning Overview: Deep Learning Book

Chapter 10: Recursive and Recurrent Nets

- Recurrent Neural Networks (RNNs)
- Unfolding Computational Graphs
- Bidirectional RNNs
- Encoder-Decoder Architectures
- Long Short-Term Memory Networks

Learning Overview: Deep Learning Book

Chapter 11: Practical Methodology

- Performance Metrics
- Baseline Models
- Hyperparameter Selection

Learning Overview: Deep Learning Book

Chapter 12: Applications

- Large-Scale Deep Learning
- Computer Vision
- Speech Recognition
- Natural Language Processing

Learning Overview: Other

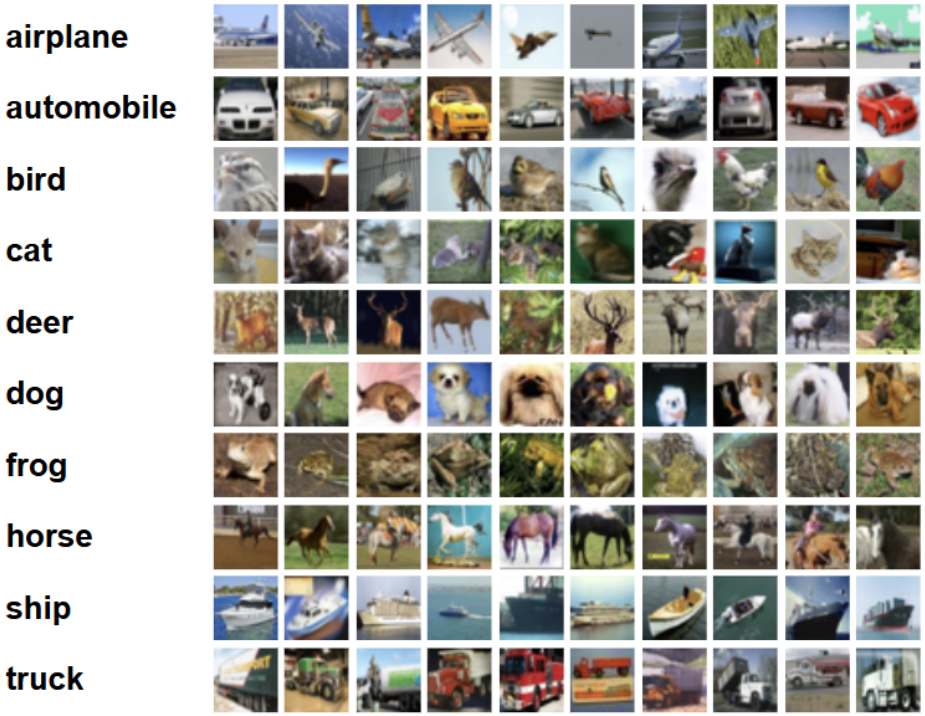
- **Papers**
  - Very Deep Convolutional Networks for Large-Scale Image Recognition
  - Efficient Post-training Quantization with FP8 Formats
  - The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks
  - ...
- **Articles**
  - Learning PyTorch: The Basic Program Structure
  - VGG-Net Architecture Explained
  - ...
- **Documentation**
  - PyTorch Documentation
  - PyTorch Quantization Documentation
  - Matplotlib Documentation
  - ...

EXPERIMENTAL OUTCOMES

Experimental Outcomes: Introduction/Motivation

- Finding a way to maintain accuracy while lowering inference time is a worthwhile endeavor as more systems use AI (Less Time $\Longrightarrow$ Less Cost).
- ChatGPT reportedly spent 2.5 million just to show the capabilities of GPT 4 on various benchmarks.
- When Deepseek entered the scene with its amazing optimizations, it sent a ripple through the stock market. A clear sign of the power of optimizations and efficient inference.

Experimental Outcomes: Datasets



CIFAR-10

ImageNet

Experimental Outcomes: Datasets

ImageNette

ImageWoof

Experimental Outcomes: Datasets

Table: Sample Counts for ImageNette Classes

Class Name	Sample Count
Tench	1350
English Springer	1350
Cassette Player	1350
Chain Saw	1244
Church	1350
French Horn	1350
Garbage Truck	1350
Gas Pump	1350
Golf Ball	1350
Parachute	1350

Experimental Outcomes: Datasets

Table: Sample Counts for ImageWoof Classes

Class Name	**Sample Count**
Shih	1350
Rhodesian Ridgeback	1350
Beagle	1350
English Foxhound	804
Border Terrier	1350
Australian Terrier	1350
Golden Retriever	1350
Old English Sheepdog	1350
Samoyed	1350
Dingo	1350

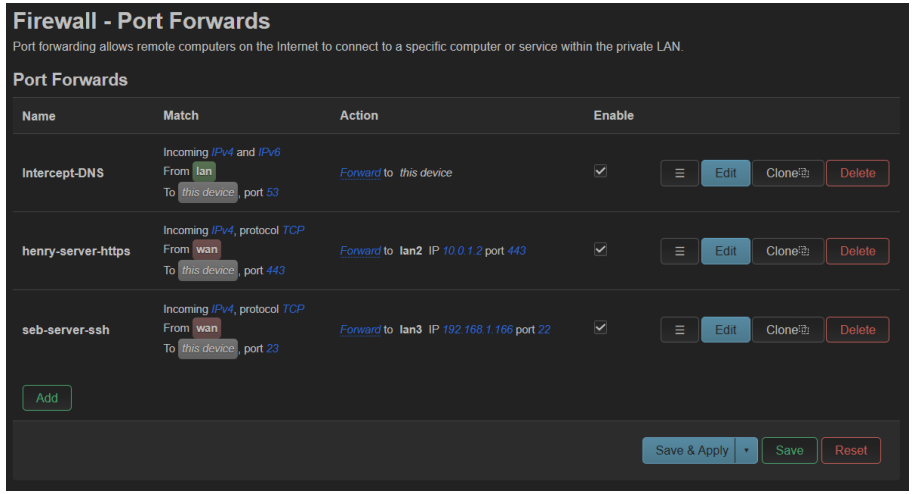
Experimental Outcomes: Datasets

Table: Sample Counts for CIFAR-10 Classes

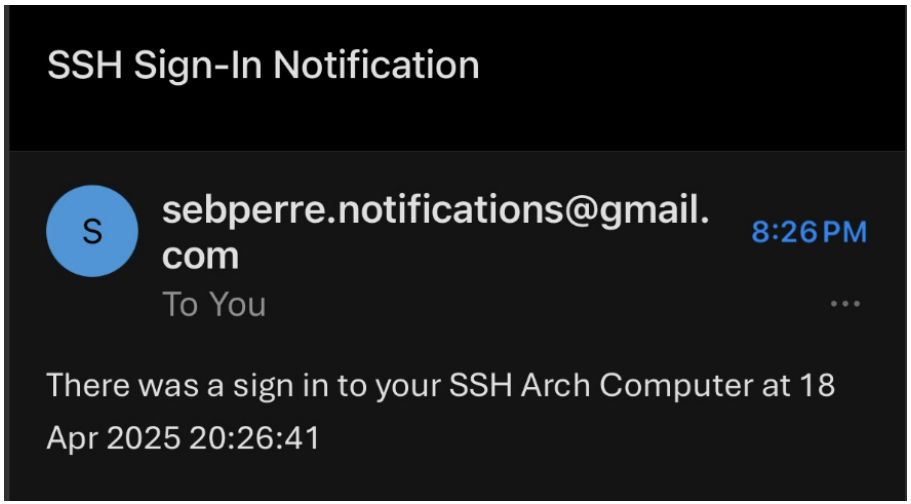
Class Name	Sample Count
Airplane	6000
Automobile	6000
Bird	6000
Cat	6000
Deer	6000
Dog	6000
Frog	6000
Horse	6000
Ship	6000
Truck	6000

Experimental Outcomes: Setting Up a Cloud GPU

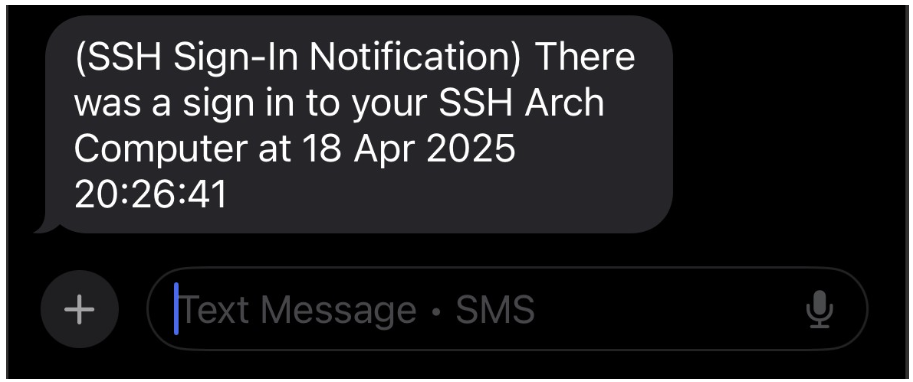
OS for Remote Desktop



Port Forwarding for External Access



Email Notification



Text Notification

Experimental Outcomes: Time Differential Between Computers and Processing Units

Table: Training Time (in seconds) for 5 Epochs. Laptop: AMD Ryzen 7 5800HS (CPU), Nvidia GeForce RTX 3050 (GPU); Remote Desktop: Intel Core i5-4460 (CPU), Nvidia GeForce GTX 1060 6GB (GPU). ImageNet, ImageNette, and ImageWoof were trained on 1000 images.

Device	**Dataset (Model)**	**CPU (s)**	**GPU (s)**
Laptop	CIFAR-10 (Simple CNN)	176.00	61.78
	ImageNet (ResNet-18)	288.30	38.44
	ImageNette (ResNet-18)	304.47	38.12
	ImageWoof (ResNet-18)	287.09	35.65
Remote Desktop	CIFAR-10 (Simple CNN)	176.00	61.78
	ImageNet (ResNet-18)	288.30	38.44
	ImageNette (ResNet-18)	336.92	45.95
	ImageWoof (ResNet-18)	333.77	44.92

Experimental Outcomes: Very Deep Convolutional Networks (VGGs)

- VGG networks are deep convolutional architectures with a simple, consistent design.
- Use small 3×3 convolutional filters stacked in sequence.
- Demonstrated that increasing depth improves image classification performance.
- Known for strong baseline performance and ease of use in research.
- Kaiming Initialization helps stabilize training by maintaining activation variance across layers.
- Especially important for deep networks with ReLU activations like VGG.

Experimental Outcomes

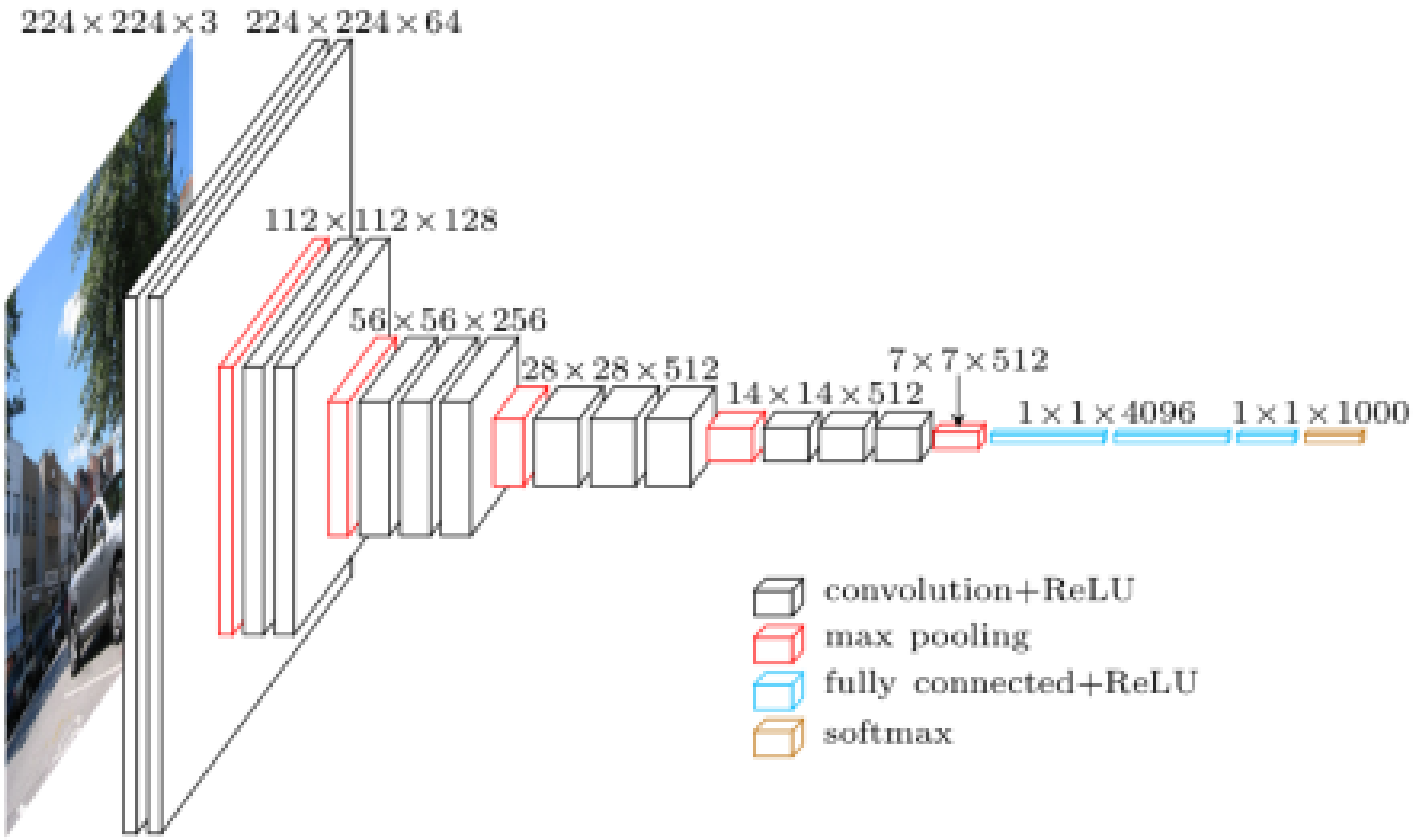


Figure: VGG Visualization

Experimental Outcomes: Testing Model Performance

CIFAR-10

Metric	Value
Training Time	4m 23s
Accuracy	0.7220
Precision	0.7261
Recall	0.7220
F1 Score	0.7218

Table: Simple CNN

Metric	Value
Training Time	22m 4s
Accuracy	0.7791
Precision	0.7989
Recall	0.7791
F1 Score	0.7837

Table: VGG-16

Experimental Outcomes: Testing Model Performance

CIFAR-10

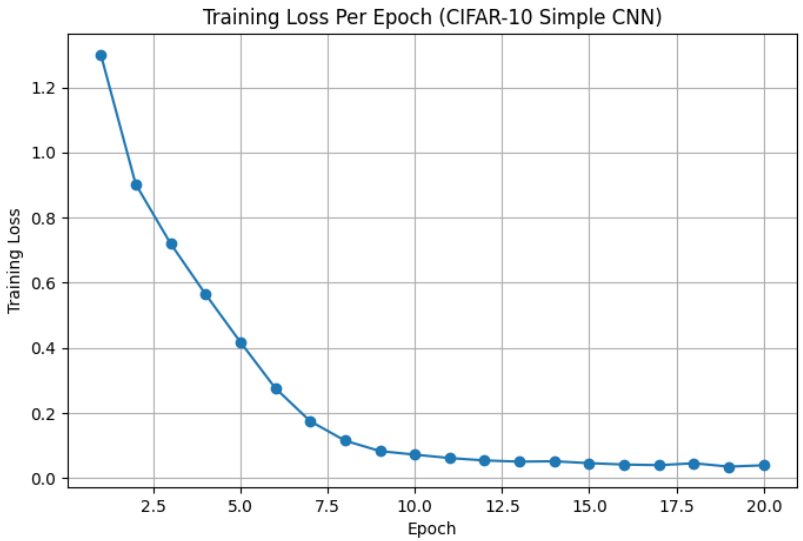


Figure: Simple CNN

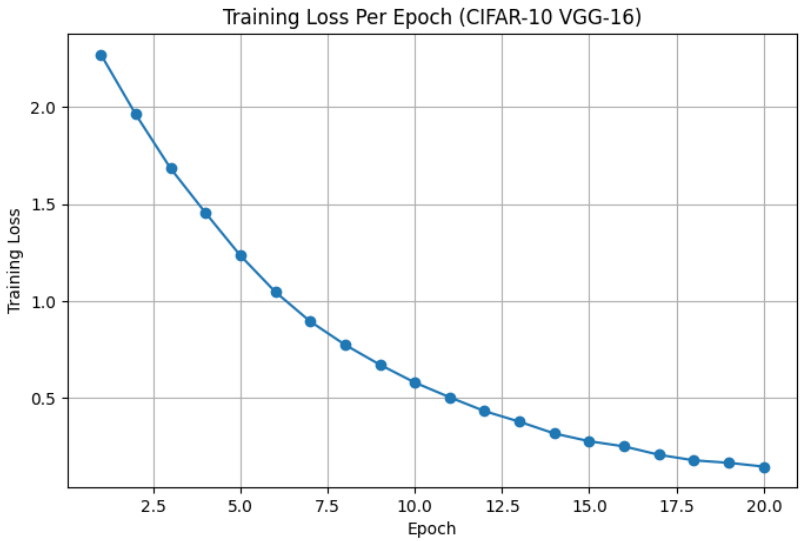


Figure: VGG-16

Experimental Outcomes: Testing Model Performance

ImageNet

Metric	Value
Training Time	30m 1s
Accuracy	0.8254
Precision	0.8349
Recall	0.8254
F1 Score	0.8271

Table: Pretrained VGG-11

Metric	Value
Training Time	35m 15s
Accuracy	0.0003
Precision	0.0003
Recall	0.0003
F1 Score	0.0003

Table: VGG-16

Metric	Value
Training Time	1h 50m 30s
Accuracy	0.0719
Precision	0.0826
Recall	0.0719
F1 Score	0.0648

Table: ResNet-18

Experimental Outcomes: Testing Model Performance

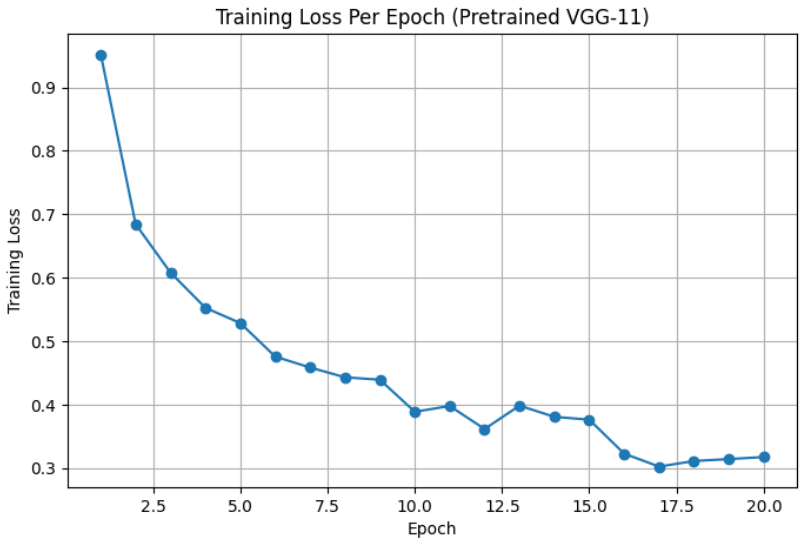


Figure: Pretrained VGG-11

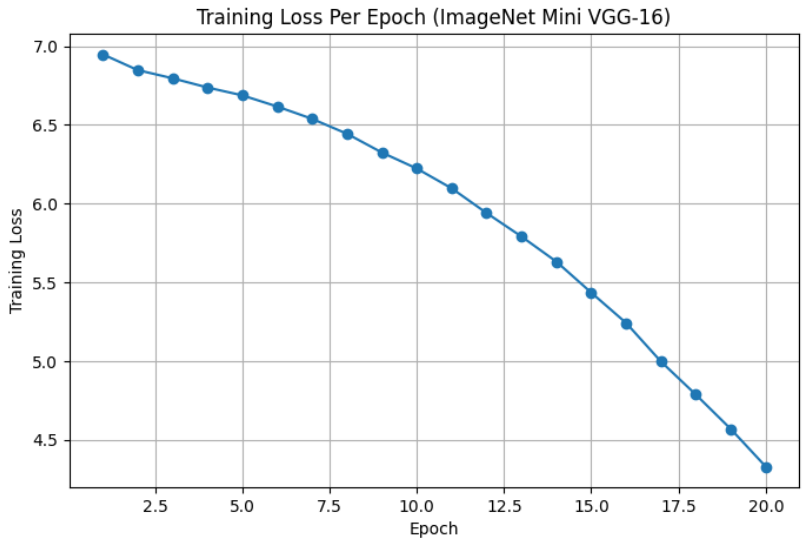


Figure: VGG-16

Experimental Outcomes: Testing Model Performance

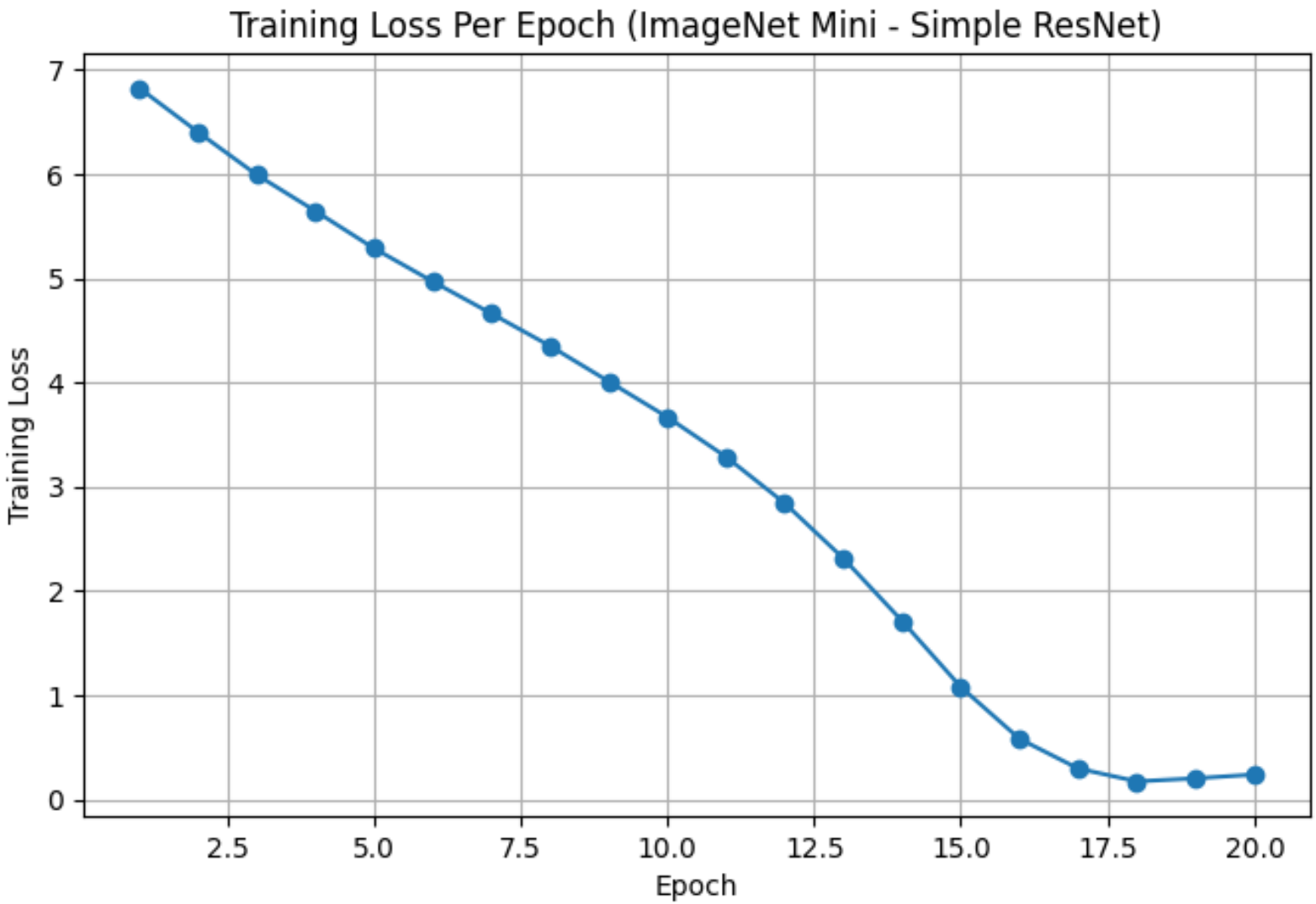


Figure: ResNet-18

Experimental Outcomes: Testing Model Performance

ImageNette

Metric	**Value**
Training Time	35m 19s
Accuracy	0.3767
Precision	0.4124
Recall	0.3767
F1 Score	0.3551

Table: VGG-16

ImageWoof

Metric	**Value**
Training Time	34m 58s
Accuracy	0.2130
Precision	0.1926
Recall	0.2130
F1 Score	0.1504

Table: VGG-16

Experimental Outcomes: Testing Model Performance

ImageNette

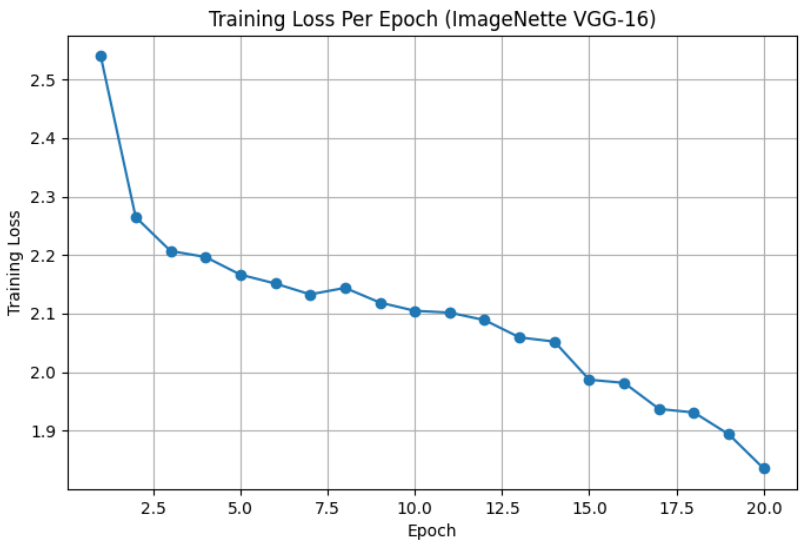


Figure: VGG-16

ImageWoof

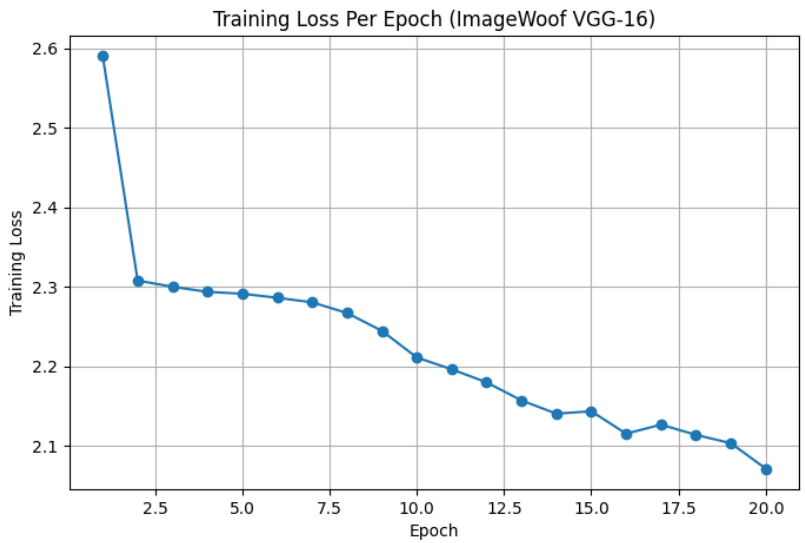


Figure: VGG-16

Motivation

Experimental Outcomes: Classification Model

Depth	Layer Configuration	# Conv	Channels per Block	Flattened
1	[64, Pool]	1	64	16,384
2	[64, 64, Pool]	2	64	16,384
3	[64, 64, Pool, 128, 128, Pool]	4	64, 128	8,192
4	[64, 64, 64, Pool, 128, 128, 128, Pool]	6	64, 128	8,192
5	[64, 64, 64, Pool, 128, 128, 128, Pool, 256, 256, 256, Pool]	9	64, 128, 256	4,096

Table: Summary of Custom VGG Architectures for CIFAR-10. Each configuration uses 3×3 convolutions with padding and ReLU activations. All models end with a classifier consisting of two hidden layers of size 512 and a final output layer for classification. Flattened feature size is taken before the first linear layer.

Experimental Outcomes: Classification Model

Oracle

Experimental Outcomes: Classification Model

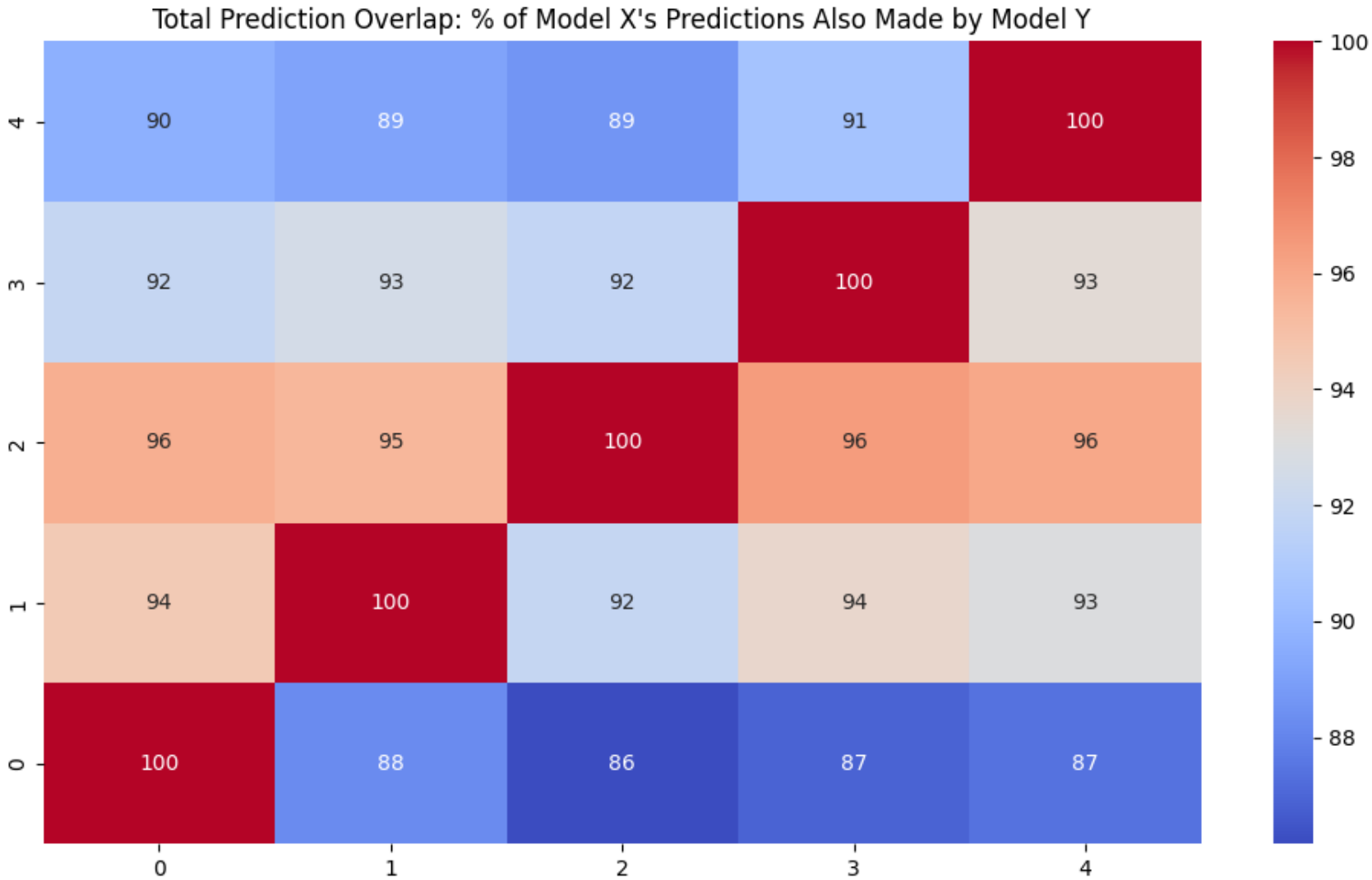


Figure: Total Prediction Overlap

Experimental Outcomes: Classification Model

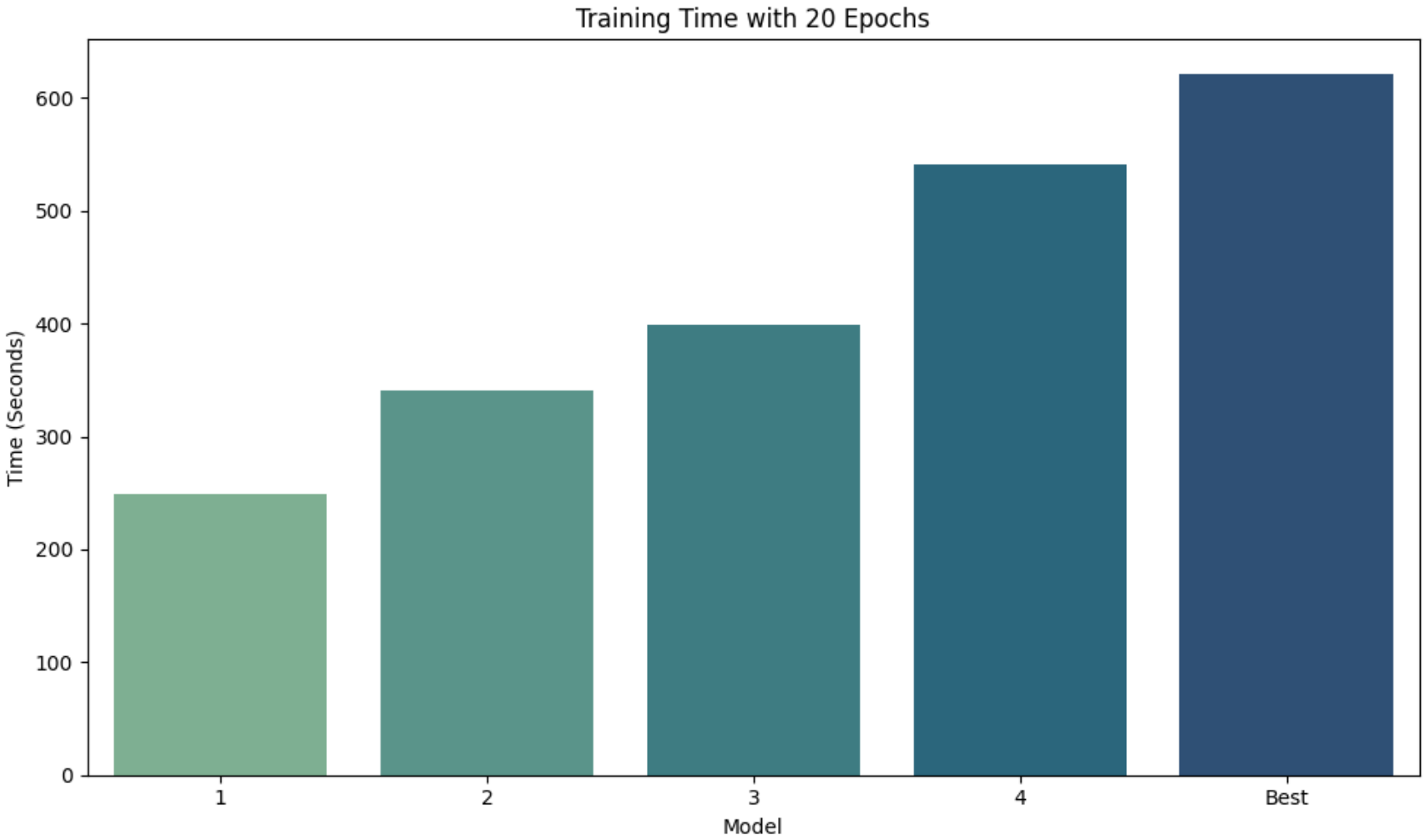


Figure: Training Time

Experimental Outcomes: Classification Model

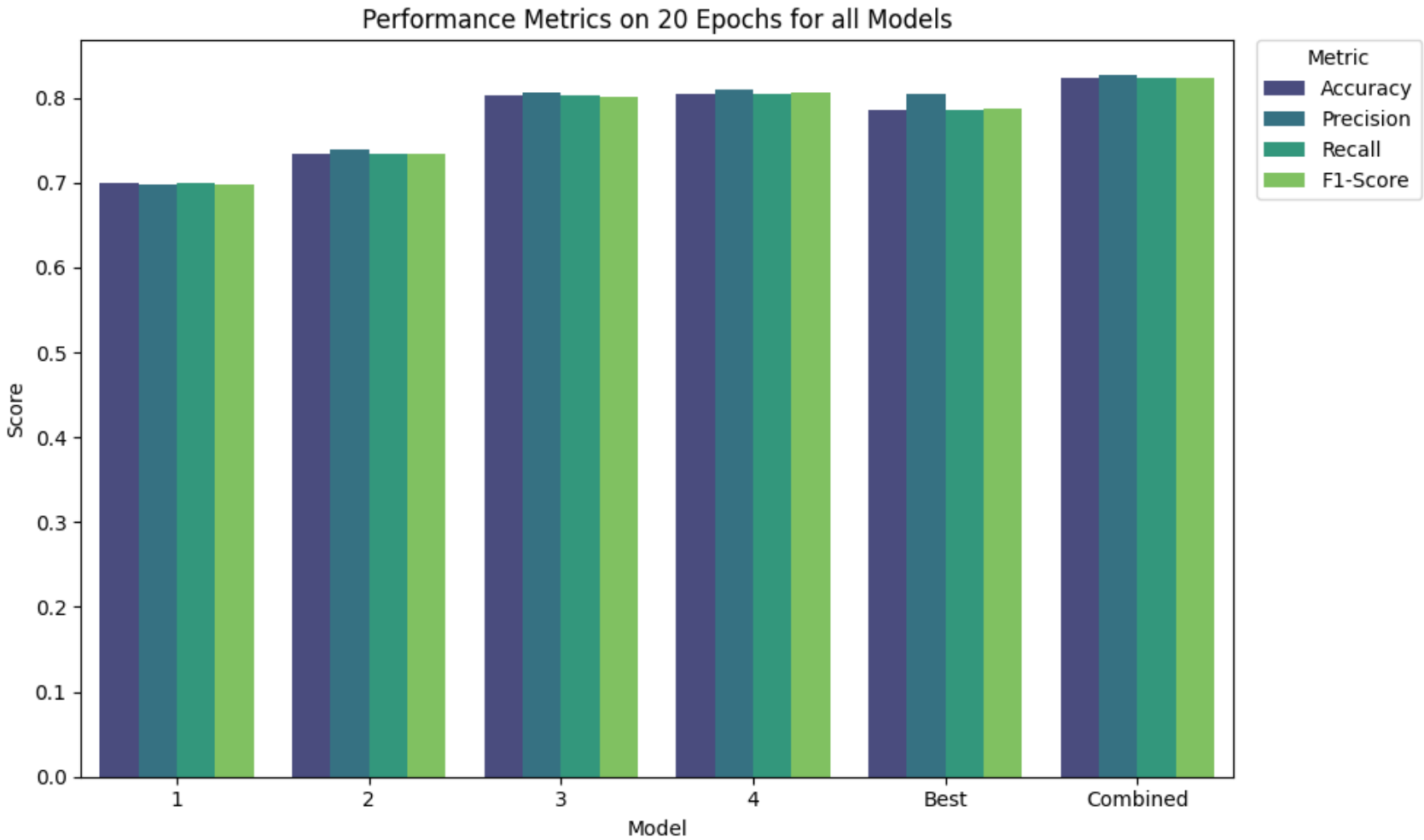


Figure: Key Performance Metrics

Experimental Outcomes: Classification Model

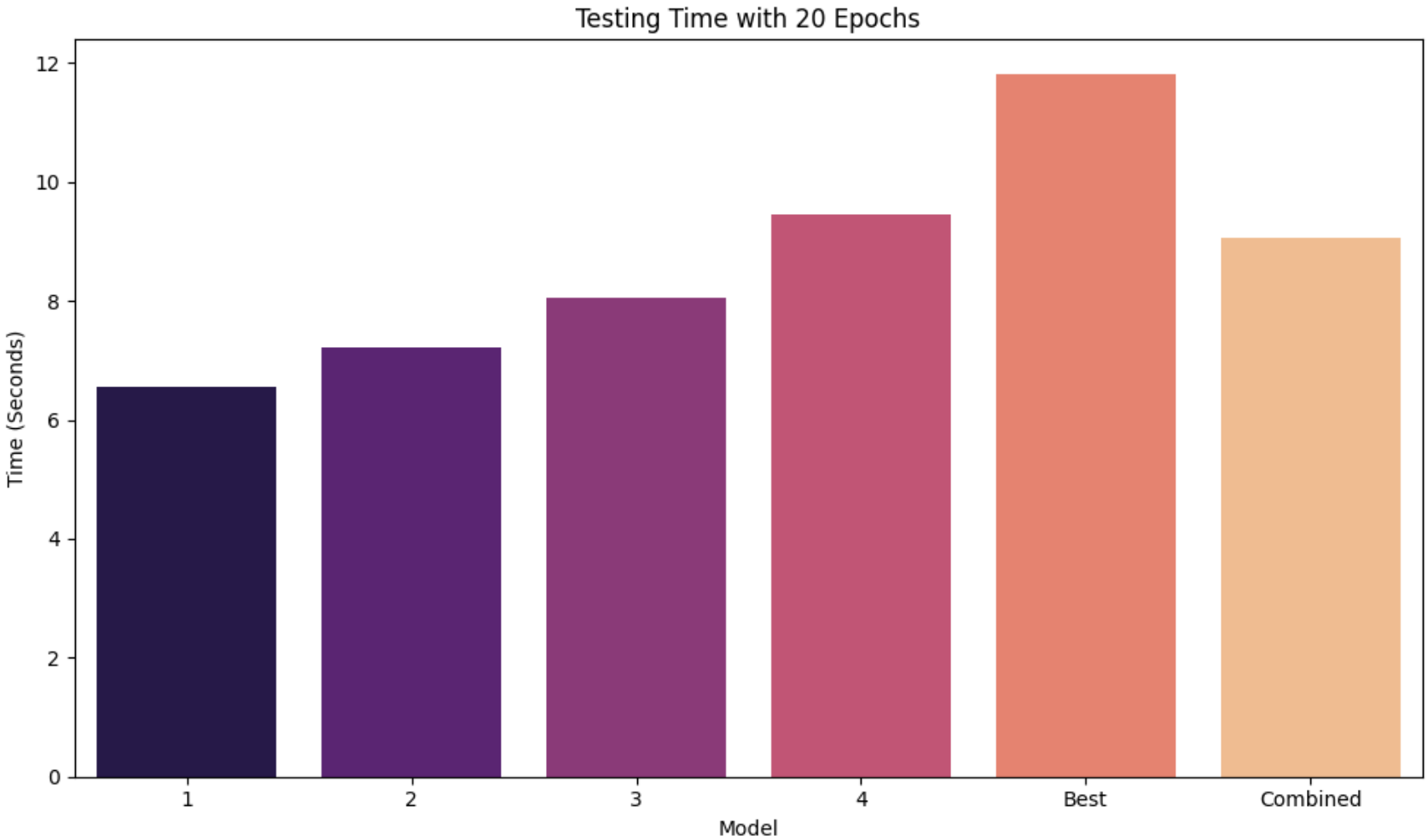


Figure: Testing Time

Efficient Classification

Experimental Outcomes: Classification Model

Statistic	**Best**	**Combined**	**Difference**
Accuracy (%)	82.38	81.45	0.93
Precision (%)	82.4832	81.3891	1.0941
Recall (%)	82.38	81.45	0.93
F1 (%)	82.37	81.3779	0.9921
Time (s)	11.359	58.8768	-47.5178

Table: Comparison of Best vs Combined Results Across Multiple Metrics

Experimental Outcomes: Quantization

Post-Training Quantization

1. Static Quantization
2. Dynamic Quantization
3. Weight-Only Quantization

Experimental Outcomes: Quantization

Other Types of Quantization

- Quantization-Aware Training
- Very-Small Quantization
- Mixed-Precision Quantization

Experimental Outcomes: Pruning

Lottery Ticket Hypothesis

- The Lottery Ticket Hypothesis proposes that inside a large neural network, there exists a smaller subnetwork that can be trained to match the full model's accuracy.
- This "winning ticket" is found by training the full model, pruning the least important weights, resetting the remaining ones, and retraining.
- It showed that we can find efficient models early on, changing the way we think about pruning and model design.

Experimental Outcomes: Pruning

1. Unstructured Pruning
    1. Magnitude-Based Pruning (L1 Norm, L2 Norm)
    2. Gradient-Based Pruning
2. Structured Pruning
    1. Filter Pruning
    2. Channel Pruning
    3. Layer Pruning
    4. Block Pruning

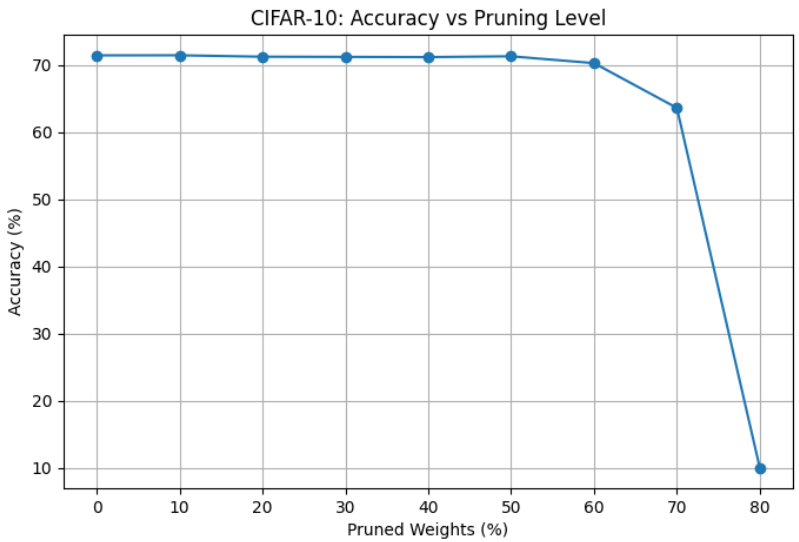
Experimental Outcomes: Pruning

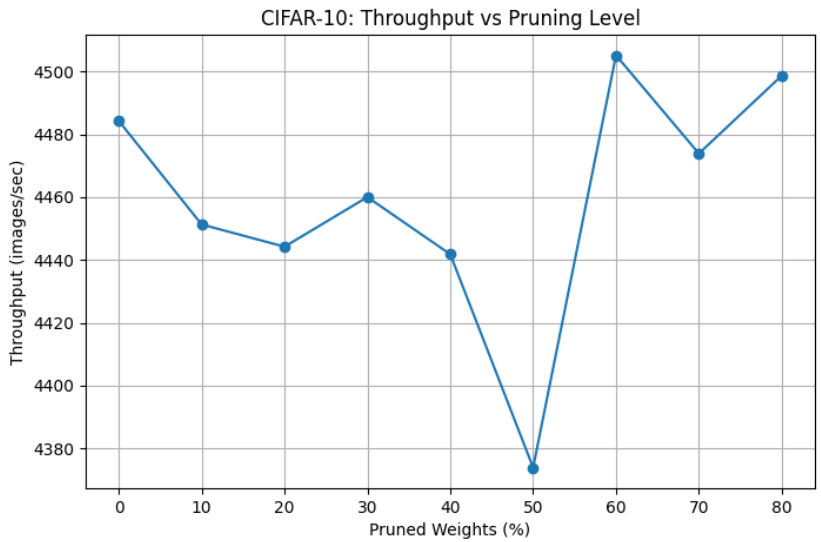
Other Types of Pruning

1. Neural Architecture Search (NAS)-based Pruning
2. Hardware-Aware Pruning
3. Dynamic/Online Pruning During Training
    1. Gradual Pruning
    2. Dynamic Sparse Training (DST)
4. Learning-Based Pruning (Trainable or Meta-Learned)
    1. L0 Regularization
    2. AutoML/Reinforcement Learning-Based Pruning
    3. Bayesian Pruning

Experimental Outcomes: Pruning

Implementation of L1 Pruning





FUTURE ENDEAVORS

Future Endeavors

Future Endeavors

- More Implementations
- Fixing the Oracle and Efficient Inference Models for ImageNette and ImageWoof
- Combining Methods
- Research Efficient Inference for Different Types of Problems and Model

Acknowledgments

I would like to thank Professor George Tzanetakis for his patience, time, and supervision.

References I

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016, ISBN: 9780262035613. [Online]. Available: https://www.deeplearningbook.org.

[2] Jeremy and the fastai community, *Imagenette: A subset of 10 easily classified classes from imagenet*, https://github.com/fastai/imagenette, 2020.

[3] A. Krizhevsky, *The cifar-10 dataset*, https://www.cs.toronto.edu/~kriz/cifar.html.

[4] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2015. arXiv: 1409.1556 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1409.1556.

[5] D. Wei, "Learning pytorch: The basic program structure," (Feb. 2024), [Online]. Available: https://medium.com/@weidagang/learning-pytorch-the-basic-program-structure-ed5723118b67.

[6] "Arch linux logos and artwork," (2025), [Online]. Available: https://archlinux.org/art/.

[7] Wikipedia contributors, "Pytorch," Wikipedia. (2025), [Online]. Available: https://en.wikipedia.org/wiki/PyTorch.

[8] Zero To Mastery, "Pytorch 101 crash course for beginners in 2025!" (Dec. 2024), [Online]. Available: https://www.youtube.com/watch?v=LyJtbe__2i0.

References II

[9] Wikipedia contributors, "Imagenet," Wikipedia. (2025), [Online]. Available: https://en.wikipedia.org/wiki/ImageNet.

[10] Stanford Vision Lab and Princeton University, "Imagenet," ImageNet. (2020), [Online]. Available: https://www.image-net.org/.

[11] I. Figotin, *Imagenet 1000 (mini)*, https://www.kaggle.com/datasets/ifigotin/imagenetmini-1000, Kaggle dataset, n.d.

[12] D. Bourke, "Zero to mastery learn pytorch for deep learning," (2025), [Online]. Available: https://www.learnpytorch.io/.

[13] Zalando Research and. collaborator, *Fashion mnist*, https://www.kaggle.com/datasets/zalando-research/fashionmnist, 2017.

[14] H. Shen, N. Mellempudi, X. He, Q. Gao, C. Wang, and M. Wang, *Efficient post-training quantization with fp8 formats*, 2024. arXiv: 2309.14592 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2309.14592.

[15] PyTorch Team, "Quantization — pytorch 2.2 documentation," (2024), [Online]. Available: https://pytorch.org/docs/stable/quantization.html.

[16] "Pytorch," (2025), [Online]. Available: https://pytorch.org/.

References III

[17] ArchWiki contributors, "Google authenticator," (2025), [Online]. Available: https://wiki.archlinux.org/title/Google_Authenticator.

[18] Arch Linux Developers, "Arch linux," (2025), [Online]. Available: https://archlinux.org/.

[19] P. Team, "Post training quantization with torch-tensorrt," (2023), [Online]. Available: https://pytorch.org/TensorRT/tutorials/ptq.html.

[20] H. Cheng, M. Zhang, and J. Q. Shi, *A survey on deep neural network pruning-taxonomy, comparison, analysis, and recommendations*, 2024. arXiv: 2308.06767 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2308.06767.

[21] J. Frankle and M. Carbin, *The lottery ticket hypothesis: Finding sparse, trainable neural networks*, 2019. arXiv: 1803.03635 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1803.03635.

[22] Y. He and L. Xiao, "Structured pruning for deep convolutional neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, 2900–2919, May 2024, ISSN: 1939-3539. DOI: 10.1109/tpami.2023.3334614. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2023.3334614.

[23] S. Vadera and S. Ameen, "Methods for pruning deep neural networks," *IEEE Access*, vol. 10, pp. 1–1, Jan. 2022. DOI: 10.1109/ACCESS.2022.3182659.

References IV

[24] GeeksforGeeks, "Vgg-net architecture explained," (Jun. 2024), [Online]. Available: https://www.geeksforgeeks.org/vgg-net-architecture-explained/.

[25] G. Boesch, "Very deep convolutional networks (vgg) essential guide," (Oct. 6, 2021), [Online]. Available: https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/.

THANK YOU FOR LISTENING!