# Role of Weight Initialization: Xavier vs He vs Random Uniform

Rahul Chhabra (25M0820)
Viraj Ashar (25M0756)
Shreyash Thok (25M0761)
Prince (25M0809)
Kamal (25M0814)

CS725: Foundation of Machine Learning, IIT Bombay
November 25, 2025

**Abstract**

This report investigates how different weight initialization strategies—naive random uniform, Xavier, and He initialization—affect the training dynamics, gradient stability, and accuracy of a CNN model trained on CIFAR-10. We compare their impact on convergence speed, gradient norms, and final validation accuracy.

## 1 Introduction

Weight initialization determines how activations and gradients propagate through a neural network during early training. Poor initialization often leads to exploding or vanishing gradients, slow convergence, or dead neurons. This study compares:

- Naive Random Uniform Initialization
- Xavier/Glorot Initialization (for Tanh)
- He/Kaiming Initialization (for ReLU)

## 2 Problem Statement

The goal is to train an identical CNN architecture on CIFAR-10 and evaluate how the three initializations influence:

- Convergence speed
- Stability of gradients and activations
- Final validation accuracy

# 3 System Setup

## 3.1 Model

A CNN with three 3x3 convolution blocks (32→64→128 channels), followed by a linear classifier. Experiments were conducted with both Tanh and ReLU activation functions.

## 3.2 Datasets

CIFAR-10 dataset consisting of 50k training and 10k validation images.

## 3.3 Hardware

Google Colab T4 GPU.

## 3.4 Framework

PyTorch.

# 4 Model Architecture

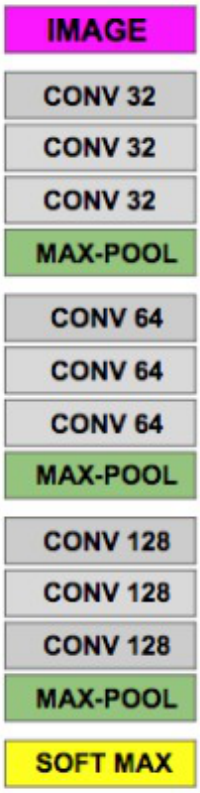



Figure shows the CNN architecture used for all experiments.

## 5 Challenges with Naive Uniform Initialization

Experiments show severe instability when using naive random uniform initialization:

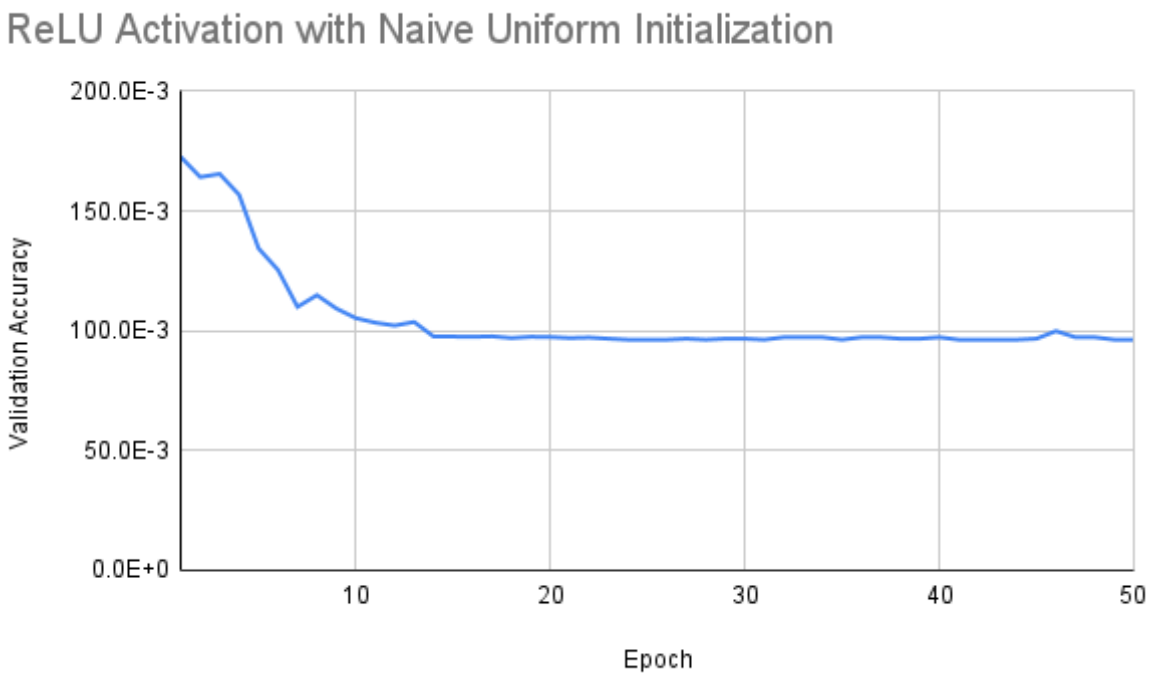
- **ReLU, bound=1**: Exploding activations and gradients; loss in millions; validation accuracy  17%.
- **ReLU, bound=0.005**: Gradients collapse to zero; model outputs constant predictions; validation accuracy  10%.
- **Tanh, bound=1**: Unstable gradients and high validation loss; accuracy  11%.
- **Tanh, bound=0.005**: Some learning happens (68.8% accuracy), but activations eventually saturate.

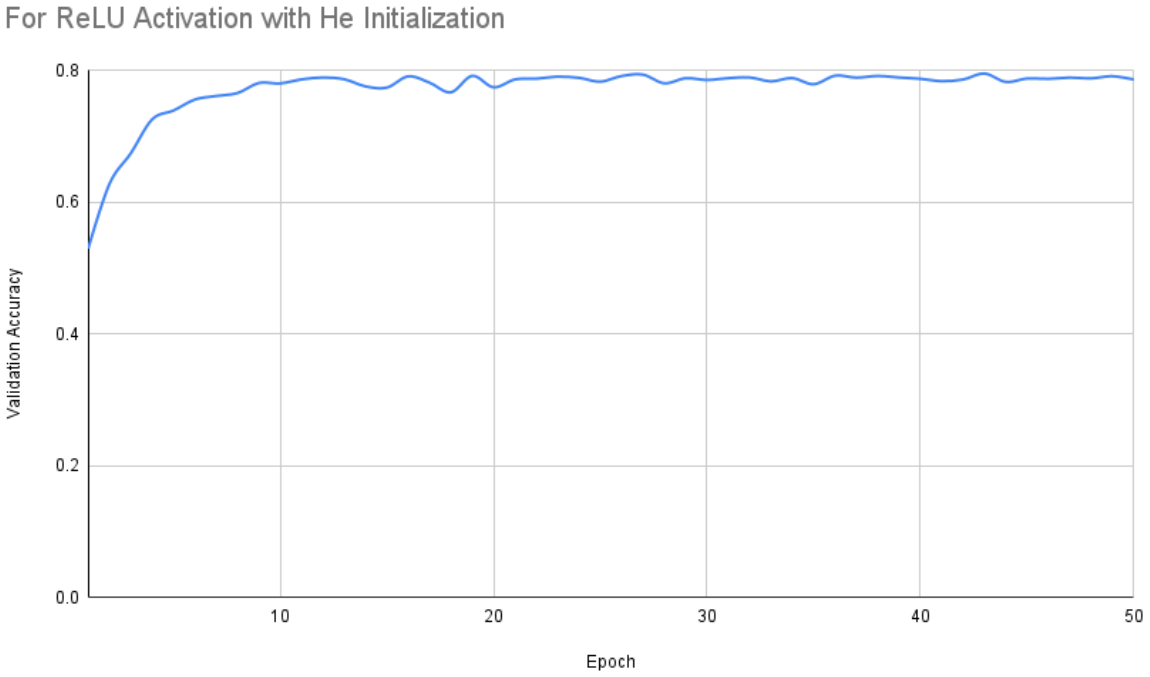
## 6 Effectiveness of Xavier and He Initialization

- **He + ReLU**: Best performance with 79.5% validation accuracy and stable gradients.
- **Xavier + Tanh**: Achieves 77.3% accuracy with smooth convergence.
- **Naive ReLU**: Either explodes or gets stuck; no meaningful learning.
- **Naive Tanh**: Slower, noisy training; saturates early.

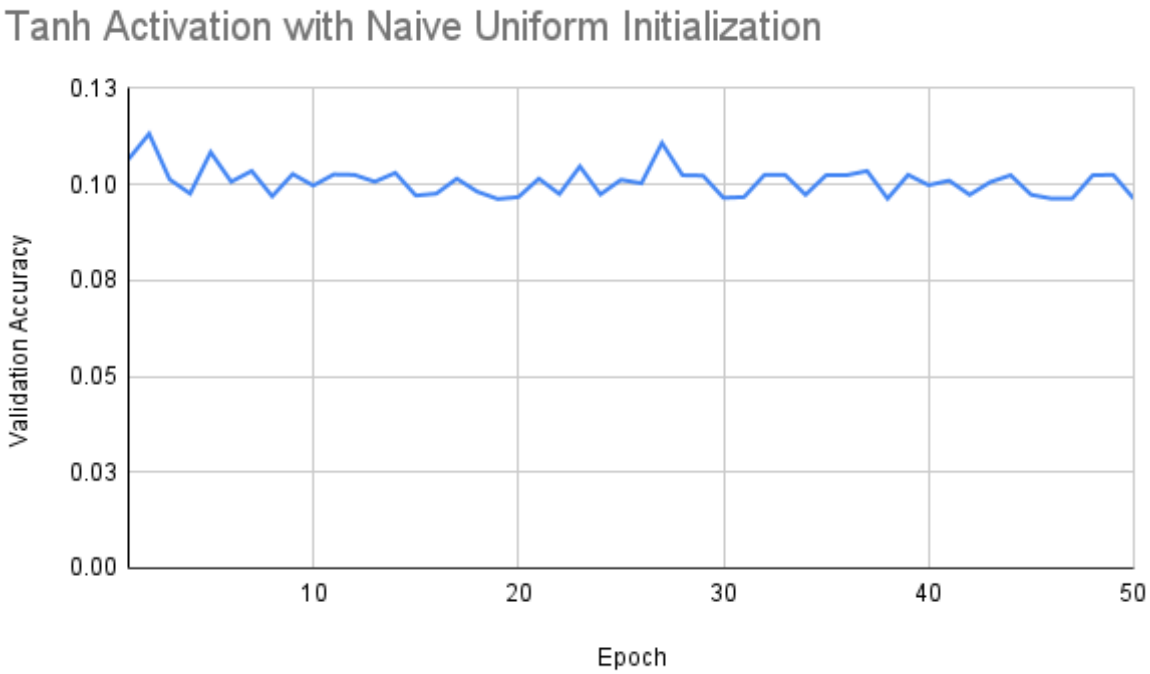
# 7 Results

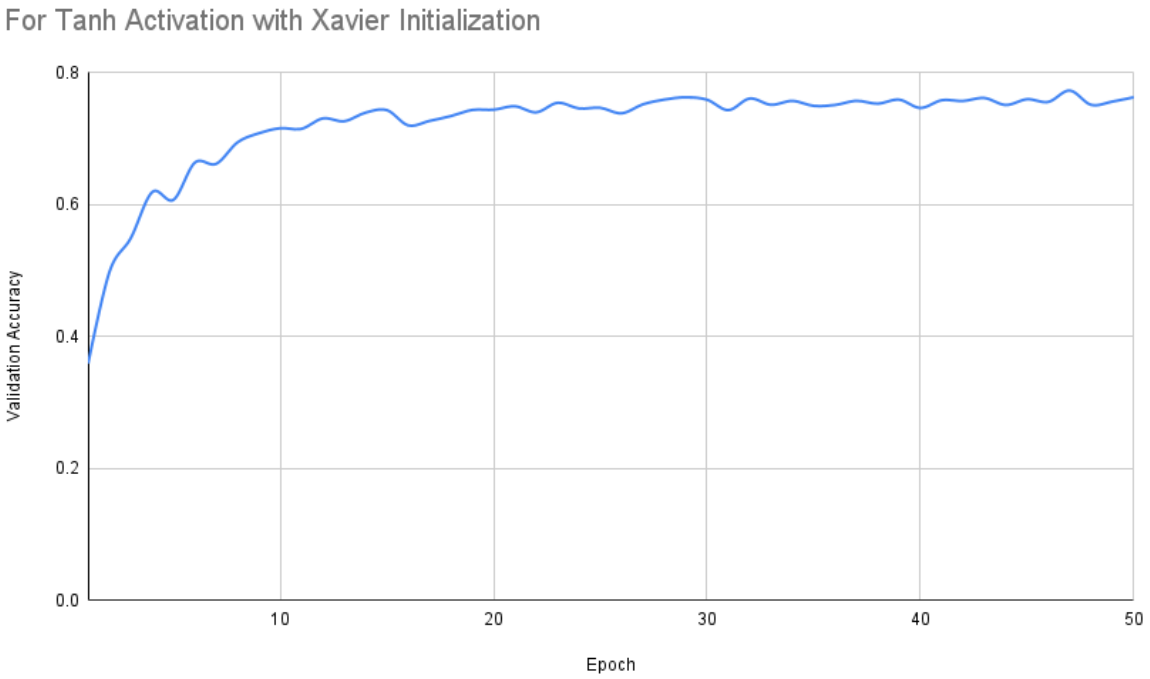
## 7.1 Validation Accuracy (ReLU)



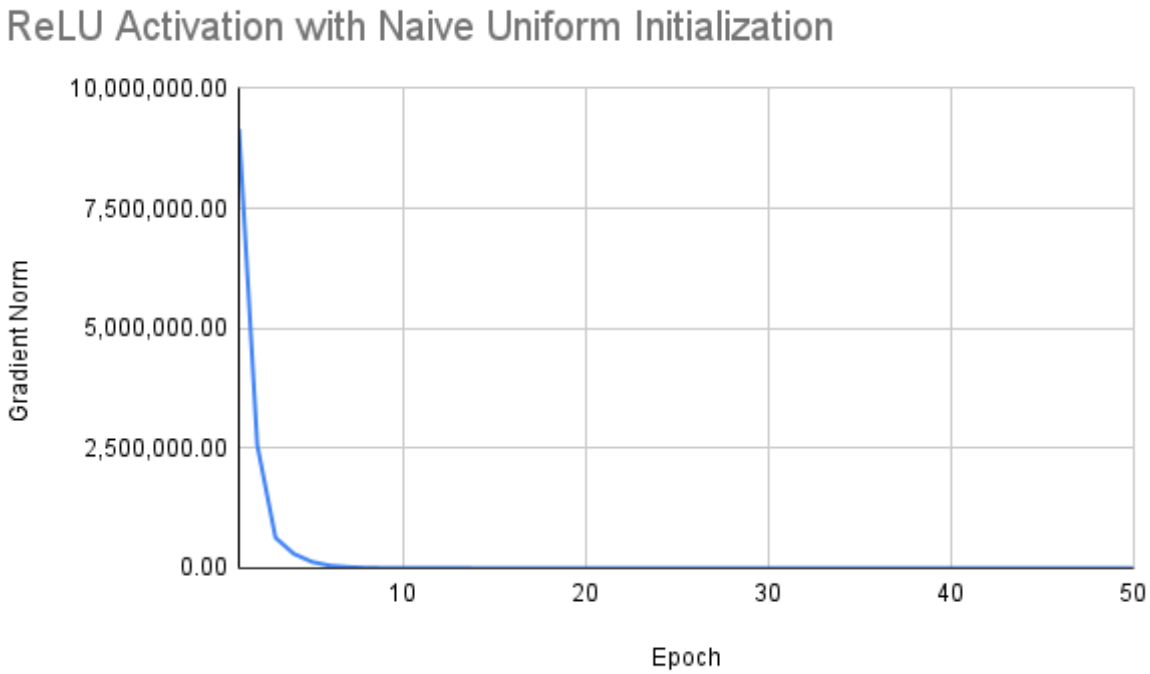



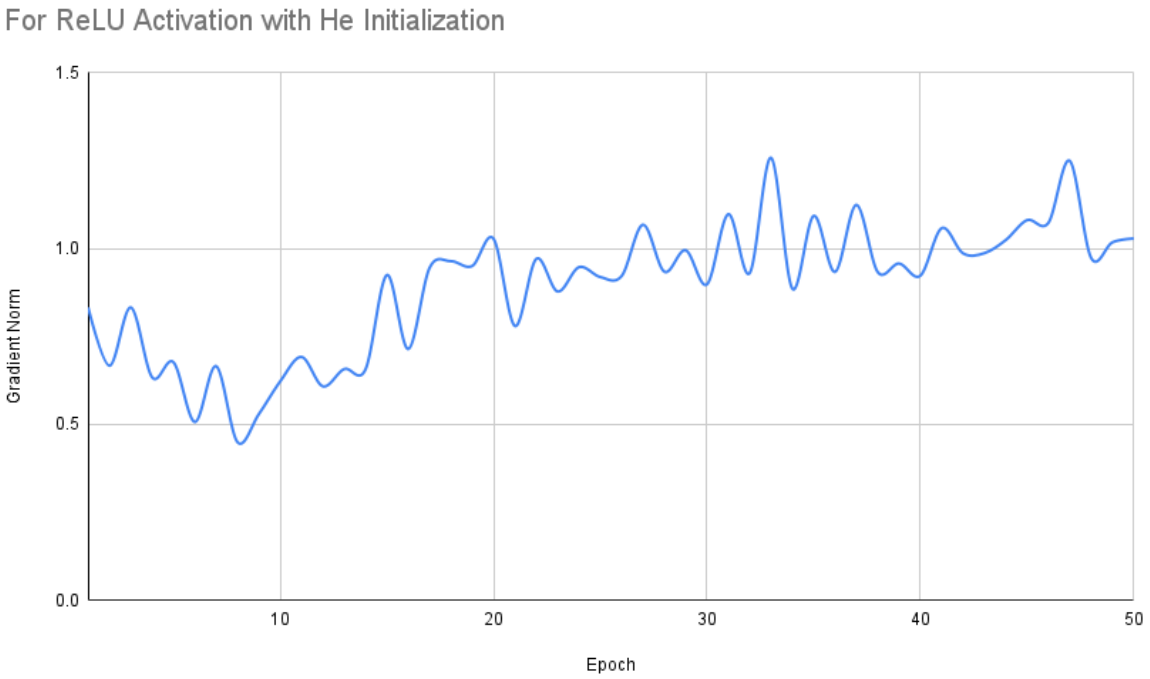
### 7.2 Validation Accuracy (Tanh)



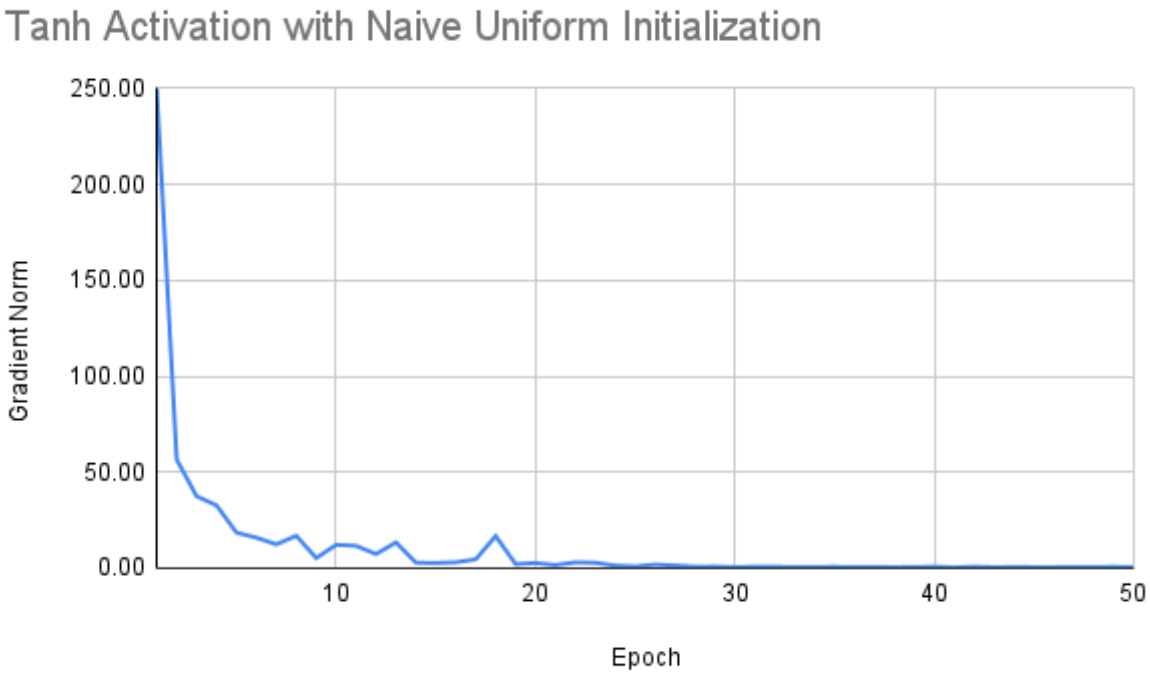



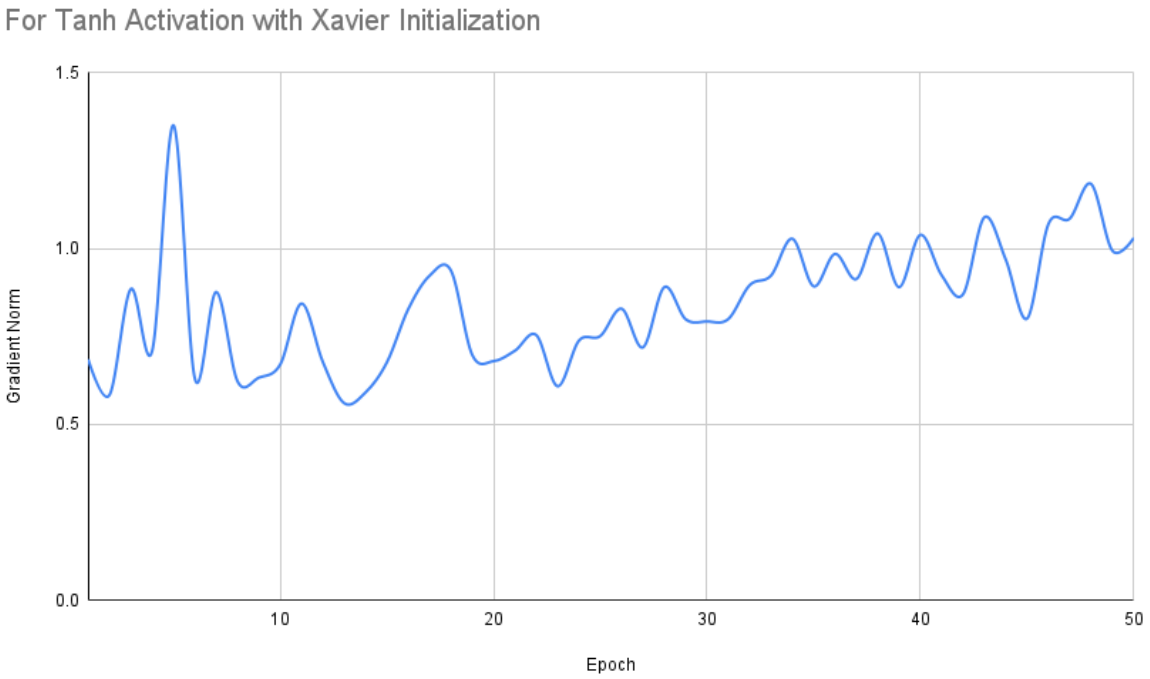
### 7.3 Gradient Norm (ReLU)



### 7.4 Gradient Norm (Tanh)



# 8 Conclusion

- He initialization provides the best combination of stability and accuracy when using ReLU (79.5% accuracy).
- Xavier initialization suits Tanh networks, achieving 77.3% accuracy.
- Naive uniform initialization is either too unstable (large bounds) or too weak (small bounds), leading to poor convergence.
- Proper initialization is essential for avoiding exploding/vanishing gradients and achieving stable learning.

# 9 Repository

**GitHub:** `https://github.com/ashar-viraj/glorot-vs-he`

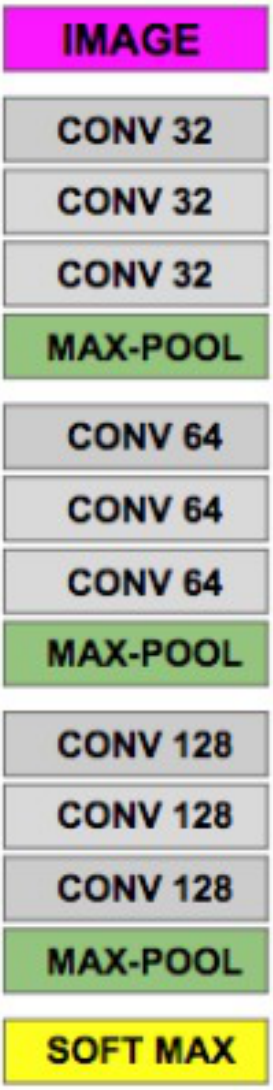



Figure 1: CNN architecture used in experiments (adapted from On Weight Initialization in Deep Neural Networks).