Overview:
- About Me
- MNIST Classification
- Assignment 1 - Keras
- Assignment 2 - PyTorch
- Research Project
About Me

**School:** Case Western Reserve University

**Major:** Data Science and Analytics

**Minor:** Economics

**Relevant Courses:**

- Algorithms
- Linear Algebra
- Statistics
- Structured & Unstructured Data

**Progress:**

- Convolutional Neural Networks
  - Keras
  - PyTorch
- Transformers
- Newton
MNIST Classification

Keras

Steps:
1. Load and pre-process data
2. Design CNN model
3. Compile and train model
4. Evaluate
5. Visualize
Evaluation Results

Final Test Accuracy: 0.973

Final Test Loss: 0.086
Assignment 1

CIFAR-100 with Keras

CIFAR-100:

- 100 classes of images
  - 600 images per class
    - 500 training
    - 100 testing

Tasks:

- Create 4 different models
- Run best model on 5 dataset sizes
Model 1: Base

Hyperparameters (unchanged for all models):
- Batch size: 128
- Epochs: ~40 (early stopping)
- Learning rate: 0.0001
- Optimizer: Adam
- Loss function: Cross Entropy
- Kernel size: 3x3

Architecture:
- 2 128x128 convolution layers
- 2 256x256 convolution layers
- 2 512x512 convolution layers
- 2x2 max pooling filter + dropout layer in between each pair of convolutional layers
- 2 fully connected layers
- Softmax activation function

Total Parameters: 6,777,060
Testing Accuracy: 52%
Time Taken: 878 s (~14.6 min)
Model 2

Architecture:
- 2 128x128 convolution layers
- 2 256x256 convolution layers
- 3 512x512 convolution layers
- 2x2 max pooling filter + dropout layer in between each pair of convolutional layers
- 2 fully connected layers
- Softmax activation function

Changes Made:
- Extra 512x512 convolution + dropout layers
- Activation: ReLU → ELU

Total Parameters: 9,136,868
Testing Accuracy: 52%
Time Taken: 718 s (~12 min)
Model 3 (best performance)

Architecture:
- 2 128x128 convolution layers
- 2 256x256 convolution layers
- 3 512x512 convolution layers
- 2x2 max pooling + dropout + batch normalization + activation in between each pair of convolutional layers
- 2 fully connected layers
- Softmax activation function

Changes Made:
- Added batch normalization + activation layers
- Added more padding

Total Parameters: 7,236,068
Testing Accuracy: 62%
Time Taken: 1898 s (~31.6 min)
Model 4

Architecture:

- 2 128x128 convolution layers
- 2 256x256 convolution layers
- 3 512x512 convolution layers
- 2x2 max pooling filter + dropout layer in between some pairs of convolutional layers
- 2 fully connected layers
- Softmax activation function

Changes Made:

- Removed batch normalization
- Reduced amount of max pooling + dropout

Total Parameters: 15,089,124

Testing Accuracy: 47%

Time Taken: 1703 s (~28.4 min)
Model 3 - Varying Dataset Size

*added data augmentation → significant improvement in performance
  - Rotations
  - Shifts
  - Flips

Size 1: 500 images per class
  - Accuracy: 69%
  - Time taken: 5051s (~1.4hr)

Size 2: 485 images per class
  - Accuracy: 68%
  - Time taken: 3827s (~1.06hr)

Size 3: 470 images per class
  - Accuracy: 68%
  - Time taken: 3911s (~1.09hr)

Size 4: 455 images per class
  - Accuracy: 66%
  - Time taken: 3213s (~53.6min)

Size 5: 440 images per class
  - Accuracy: 67%
  - Time taken: 3750s (~1.04hr)
Insights

- More data → more time to train
- More data → more accuracy
- More convolution layers → more accuracy
  - Excessive amount could result in overfitting
- Excessive max pooling / insufficient padding → less accuracy
- Batch normalization → more accuracy
Assignment 2

CIFAR-100 with PyTorch

CIFAR-100:
- 100 classes of images
  - 600 images per class
    - 500 training
    - 100 testing

Tasks:
- Run VGG16 model on CIFAR-10
- Run VGG16 model on CIFAR-100
VGG16 on CIFAR-10

Testing Accuracy: 78.4%

Testing Loss: 0.695
VGG16 on CIFAR-100

Testing Accuracy: 21.7%

Testing Loss: 3.22
Research Project

Visual-LLM In-Context Learning

Mentor: Dr. Ser-Nam Lim

Objective:

- Classify content within image
- Perform visual in-context learning → generate descriptions of identified content
First Steps

1. Get CLIP up and running
2. Curate database of texts to be used for CLIP
3. Get LLaMA and LLaVA models running

Current Progress:

- Familiarizing myself with codebases and papers for:
  - CLIP
    - Generates text based on image
  - BLIP2
    - Pre-training strategy
  - LLaMA
    - Student-teacher model for language
  - LLaVA
    - Large language and vision assistant
Thank you!

Questions?