JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Data Science: Artificial Intelligence (BU.920.624)

Group Assignment #1

Due on November 6th via Canvas

Group Members: ______________________________________

1. **AI Model Development Using No-Code Tools for Image Classification**

   In this exercise, you will develop an AI model for image classification using a no-code AI development tool such as Google Teachable Machine, Roboflow, or a Generative AI tool. The purpose of this task is to give you hands-on experience with AI tools and to demonstrate how AI can be leveraged for practical business applications:

   - **Step 1:** Choose a problem that can be solved through image classification. For example, you could classify objects (fruits, office supplies, etc.) or detect facial expressions (happy, sad, neutral). Briefly explain the problem you selected and why it's relevant in a business context.

   - **Step 2:** Collect or download a small image dataset to train your AI model. You can gather images manually via image search or by taking photos with your phone, or use a publicly available dataset from platforms like Roboflow, Kaggle, or Github. Briefly describe your dataset, specifying the classes (labels) and the number of images per class.

   - **Step 3:** Train the model using your chosen tool and provide a screenshot of the model and the training process. Report on the model's prediction accuracy and how it performed on the test data.

   - **Step 4:** Reflect on the results. How well did the model perform? Were there any challenges in training or obtaining accurate predictions? How could this model be applied in a real-world business scenario?

Intention:

- **Translate a real business problem into an AI formulation**

  - You had to move from a vague operational issue ("who is allowed on this road?") → a **formal multi-class classification task**.

- **Understand the full AI lifecycle without heavy coding**

  - Data collection
  - Label definition
  - Training
  - Testing
  - Performance evaluation
  - Real-world deployment logic

- **Experience real-world data challenges**

  - Viewpoint variation
  - Lighting differences
  - Occlusion

- Camera angle bias
- Dataset representativeness

- **Connect AI outputs to economic and operational consequences**

  - Your business framing around **automated enforcement and fines** directly aligned with:

    “transform unstructured data into tools with business and human value”

2. **Understanding Linearly Classifiable vs. Non-Linearly Classifiable Datasets.**

   What is a linearly classifiable dataset? What is a non-linearly-classifiable dataset? Provide an example dataset that is not linearly classifiable. You may plot your dataset without listing each data point.

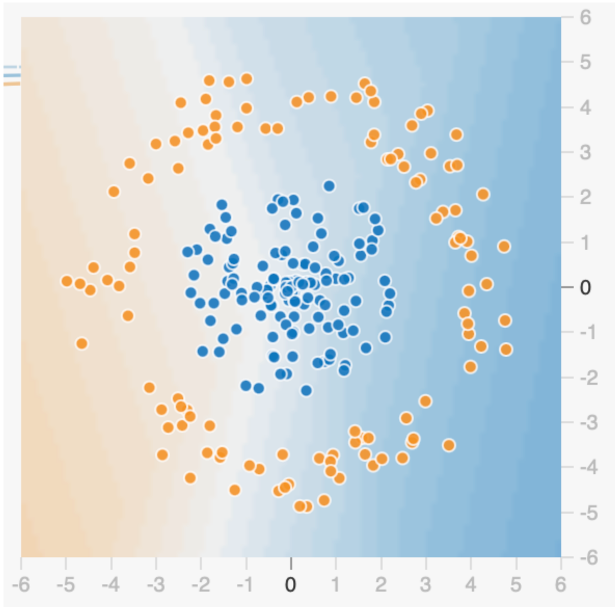
   Intention:

This question exists to ensure you **grasp why neural networks are even necessary.** Specifically, you were meant to learn:

1. **The limitation of linear models**
   - Logistic regression
   - Linear SVM
   - Single-layer perceptron
2. **Why feature engineering or hidden layers are required**
   - If a dataset is not linearly separable, **no straight decision boundary will ever work**, no matter how much data you add.
3. **Geometric intuition behind classification**
   - Decision boundaries as **hyperplanes**
   - Non-linear manifolds in real data
4. **Why deep learning exploded after 2012**
   - Because most real-world data (images, speech, behavior) is **fundamentally non-linear**

3. **Building and Describing a Neural Network for Non-Linear Classification.**

   Consider the following training dataset from TensorFlow Playground (https://bit.ly/annga1):



   The orange points are labeled with class label 0, and the blue points are labeled with class label 1. Using the TensorFlow Playground tool (see the link above), train a feedforward neuron network model. Write a mathematical expression using linear layers and ReLU activations that represents the structure of your model.

   Intention:

This is the **most conceptually important question in the assignment.** It was designed to force you to understand:

1. **A neural network is just stacked linear algebra**

$$x \to W_1 x + b_1 \to ReLU \to W_2 h + b_2 \to ReLU \to W_3 h + b_3$$

2. **ReLU is what makes non-linear separation possible**
   - Without ReLU, the entire network collapses into **one giant linear function**
3. **Hidden layer width controls model capacity**
   - Your change from:
     - 4 neurons ⟶ 3 neurons
     - Directly changed the **decision surface complexity**
4. **Loss reduction reflects actual geometric fitting**
   - Going from **0.516** ⟶ **0.001** is not cosmetic — it proves the model learned the non-linear structure.
5. **Mathematical accountability**
   - The professor wanted you to be able to:
     - Translate visuals
     - Into equations

- With explicit **weight matrices and bias vectors**

4. **Convolutional Filter Application.**

   A two-dimensional, 3 × 3 convolutional filter, that is,

1	0	1
0	1	0
0	0	1

   is applied to the following two-dimensional 5 × 5 input feature map

0	1	1	1	0
0	1	0	0.5	1
0	0	1	0	1
0	0	0.5	1	1
0	0	0	0	0

   What is the shape of the output feature map? Represent the output feature map.

Intention:

This question was designed to force **mechanical fluency** with CNNs — not conceptual hand-waving.

You were being trained to understand:

1. **How spatial features are numerically extracted**
   - A CNN does not "see" objects.
   - It computes **weighted local dot products**.
2. **How output shapes are determined**

$$(n - f + 1) \times (n - f + 1)$$

   This is **critical** for:

   - Memory planning
   - Architecture design
   - Feature map stacking

3. **Why CNNs preserve spatial locality**
    - You literally saw:
        - Which pixels influence which outputs
4. **That deep learning relies on extremely simple arithmetic**
    - Multiplication
    - Addition
    - Sliding windows