# Deep Learning Assignment

Name: Shabiqun Nahar

ID: 22308006

## Part A:

**Activation Functions:** activation functions are mathematical function used in neural networks to introduce non-linearity, enabling the model to learn complex patterns. In deep learning, activation functions are crucial components in neural networks. They define how the weighted sum of inputs and biases (the raw output from a Neuron) is transformed into the neuron's final output. This Transformation introduces non linearity into the network, allowing it to learn complex mappings from input to output

- **Sigmoid Activation Function**
  **Formula:** $f(x) = \frac{1}{1+e^{-x}}$
  Use Case: Binary classification problems.
  **Properties:** Output Range: (0, 1)
  Shape: S-shape curve (Logistic function).
  **Advantages:** Smooth gradients, useful for will probabilistic outputs. Works well in the output layer binary classification tasks.$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
  **Limitations:** Vanishing gradient for large inputs.

- **Tanh Activation Function**
  **Formula:** Type equation here.
  Use Case: Centered data around zero.
  **Properties:** Output Range: (-1, 1)
  Shape: S-shape curve, like sigmoid but zero centered.
  Advantages: Outputs are zero-centered, which with faster optimization in hidden layers.
  Batter than sigmoid for hidden layers due to its broader rangehelps
- **ReLU( Rectified Linear Unit)**
  **Formula:** f(x) = max(0,x)
  Use Case: Hidden layers in deep networks.
  **Properties:** Output range: [0, ∞)
  Computationally efficient and simple.
  **Advantages:** Avoids the vanishing gradient problem for positive velues.
  Highly efficient for deep neural and widely used in hidden layer.
  **Limitations:** Dead neurons for negative inputs.
- **Leaky ReLU**
  **Formula:** $f(x) = x \; if \; x > 0, else \; f(x) = ax$

Use Case: Solves ReLU's dead neuron problem.
**Properties:** Output Range: (-∞, ∞)
Allows small negative outputs to avoid inactive neurons.
**Advantages:** Solves the dying ReLU problem by allowing a small gradient for negative inputs.
**Limitations:** The choice of alpha can affect model performance and may require turning.

**2. Optimization Algorithms**

- **Stochastic Gradient Descent (SGD):**
  - Updates weights using a small random batch.
  - **Pros:** Simplicity, robustness.
  - **Cons:** Slow convergence, sensitive to learning rate.
  - **Use Case:** Suitable for large scale dataset.

- **Adam (Adaptive Moment Estimation):**
  - Combines momentum and adaptive learning rates.
  - **Pros:** Faster convergence, work well with sparse gradient.
  - **Cons:** Requires more memory, turning needed.
  - **Use Case:** Works well in tasks requiring fast convergence and sparse gradients

- **RMSProp:**
  - Divides learning rate by an exponentially decayingaverage of squired gradients.
  - **Pros:** Effective for recurrent networks.
  - **Cons**: Sensitive to hyperparameters.
  - **Use Case:** Commonly used for reccurent neural network (RNNs).

# Part B:

- Problem Understanding: Build a Convolution Neural Network for image classification using CIFAR-10 dataset.
- Model Design: 3 convolution layers, 2 fully connected layers, ReLU activation, Normalization and dropout.
- Result: training included, Test accuracy: <Placeholder for test accuracy>, confission matrix attached.

**CONCLUSION:** The designed CNN effectively classifies images in the CIFAR-10 dataset with competitive accuracy. While some errors persist due to class similarities and data limitations, these can be mitigated by using more advanced architectures.Type equation here.