**University of Central Missouri**

**Department of Computer Science & Cybersecurity**

**CS5720 Neural network and Deep learning**

**Spring 2025**

**Home Assignment 1.**

**Student name: Sushma Veluru  
 700765390**

**Submission Requirements:**

* Once finished your assignment push your source code to your repo (GitHub) and explain the work through the ReadMe file properly. Make sure you add your student info in the ReadMe file.
* Submit your GitHub link and video on the BB.
* Comment your code appropriately ***IMPORTANT.***
* Make a simple video about 2 to 3 minutes which includes demonstration of your home assignment and explanation of code snippets.
* Any submission after provided deadline is considered as a late submission.

1. **Tensor Manipulations & Reshaping**

**Task: Tensor Reshaping & Operations**

1. Create a random tensor of shape (4, 6).
2. Find its rank and shape using TensorFlow functions.
3. Reshape it into (2, 3, 4) and transpose it to (3, 2, 4).
4. Broadcast a smaller tensor (1, 4) to match the larger tensor and add them.
5. Explain how broadcasting works in TensorFlow.

**Expected Output:**

1. Print rank and shape of the tensor before and after reshaping/transposing.
2. **Loss Functions & Hyperparameter Tuning**

**Task: Implement and Compare Loss Functions**

1. Define true values (y\_true) and model predictions (y\_pred).
2. Compute Mean Squared Error (MSE) and Categorical Cross-Entropy (CCE) losses.
3. Modify predictions slightly and check how loss values change.
4. Plot loss function values using Matplotlib.

**Expected Output:**

* Loss values printed for different predictions.
* Bar chart comparing MSE and Cross-Entropy Loss.

1. **Train a Model with Different Optimizers**

**Task: Train MNIST Model with Adam & SGD**

1. Load the MNIST dataset.
2. Train two models: One with Adam and another with SGD.
3. Compare training and validation accuracy trends.

**Expected Output:**

* Accuracy plots comparing **Adam vs. SGD** performance.

1. **Train a Neural Network and Log to TensorBoard**

**Task Description**

1. Load the MNIST dataset and preprocess it.
2. Train a simple neural network model and enable TensorBoard logging.
3. Launch TensorBoard and analyze loss and accuracy trends.

**Expected Outcome:**

* The model should train **for 5 epochs** and store logs in the "logs/fit/" directory.
* You should be able to visualize **training vs. validation accuracy and loss** in TensorBoard.

**4.1 Questions to Answer:**

1. What patterns do you observe in the **training and validation accuracy curves**?

* Observed Patterns in Training and Validation Accuracy Curves:

1. Typically, training accuracy increases steadily as the model learns from the training data.
2. Validation accuracy may also improve but might plateau or fluctuate due to generalization limits.
3. If validation accuracy diverges significantly from training accuracy, it may indicate overfitting.
4. How can you use **TensorBoard to detect overfitting**?

* Using TensorBoard to Detect Overfitting:

1. In TensorBoard, check the accuracy and loss plots. Overfitting is evident if training accuracy keeps rising while validation accuracy plateaus or drops.
2. The loss curve may show decreasing training loss while validation loss increases, further indicating overfitting.
3. Monitoring weight distributions and histograms can also help detect overfitting.
4. What happens when you increase the number of epochs?

* Effect of Increasing the Number of Epochs:

1. Initially, both training and validation accuracy improve.
2. After a certain point, training accuracy may continue to increase, while validation accuracy stagnates or drops, showing overfitting.
3. Increasing epochs without proper regularization (dropout, weight decay) might lead to a model that performs well on training data but poorly on unseen data.