**University of Central Missouri**

**Department of Computer Science & Cybersecurity**

**CS5720 Neural Networks and Deep Learning**

**Summer 2025**

**Home Assignment 3. (Cover Ch 7)**

**Student name: John Weis**

**Submission Requirements:**

* Total Points: 100
* 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 BrightSpace.
* 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.

**Q1: Implementing an RNN for Text Generation**

**Task:** Recurrent Neural Networks (RNNs) can generate sequences of text. You will train an **LSTM-based RNN** to predict the next character in a given text dataset.

1. Load a **text dataset** (e.g., "Shakespeare Sonnets", "The Little Prince").
2. Convert text into a **sequence of characters** (one-hot encoding or embeddings).
3. Define an **RNN model** using LSTM layers to predict the next character.
4. Train the model and generate new text by **sampling characters** one at a time.
5. Explain the role of **temperature scaling** in text generation and its effect on randomness.

***Hint:*** *Use tensorflow.keras.layers.LSTM() for sequence modeling.*

**Q2: Sentiment Classification Using RNN**

**Task:** Sentiment analysis determines if a given text expresses a positive or negative emotion. You will train an **LSTM-based sentiment classifier** using the IMDB dataset.

1. Load the **IMDB sentiment dataset** (tensorflow.keras.datasets.imdb).
2. Preprocess the text data by **tokenization** and **padding** sequences.
3. Train an **LSTM-based model** to classify reviews as **positive or negative**.
4. Generate a **confusion matrix** and classification report (accuracy, precision, recall, F1-score).
5. Interpret why **precision-recall tradeoff** is important in sentiment classification.

***Hint:*** *Use confusion\_matrix and classification\_report from sklearn.metrics.*

**Q3: Comparing Optimizers – Adam vs. RMSprop vs. SGD**

**Task:** Optimizers influence training speed and accuracy. You will train a **deep learning model** using different optimizers and compare their performance.

1. Implement a **feedforward neural network** for digit classification (use the **MNIST dataset**).
2. Train the model **three times**, each time using a different optimizer (Adam, RMSprop, SGD).
3. Record the **training loss & accuracy** for each optimizer.
4. Plot the **loss curves** for all three optimizers.
5. Compare the **convergence speed & final accuracy**: Which optimizer performs best and why?

***Hint:*** *Use tensorflow.keras.optimizers.Adam(), RMSprop(), and SGD().*