# Project #4

CIS 2541 - Prof. John P. Baugh – Oakland Community College – OR

## Objectives

- To learn more about deep learning techniques
- To learn about Keras and Tensorflow

## Instructions

You may work in teams of up to **4 people**.

This project includes both a relatively simple implementation portion, and a short essay.

### Part 1: Essay / Short Answers

You must research each of the following, and write a short summary/essay about what they do, how they work, etc. You should cite your references near the end of your essay. You should separate your essay by number so I know where to look for the answers to each:

1. Discuss / write about what an activation function is, and how they are used in deep learning/neural networks.

2. In particular, write about both of the following two activation functions (how do they work, when are they used):
   a. Softmax
   b. Rectified Linear Unity (ReLU)

3. Pick **any two** of the following **activation functions** and research them, and then write a summary of what each does and how they are different.

- Sigmoid (Logistic)
- Hyperbolic Tangent (tanh)
- Leaky ReLU
- Parametric ReLU
- Exponential Linear Unity (ELU)
- Swish

- Softplus

4. What is a **Sequential** model in the context of Keras?

5. Write about each of the following layer types that can be used with neural networks:
   a. Dense
   b. Flatten

Save your essay/short answers with citations at the end, in a PDF, and turn it in inside the same folder containing your implementation (Part 2).

## Part 2: Implementation

There are **five** TODOs that you must provide the code for. The rest of the code should work. You may have to make minor modifications, but once you properly provide the TODO portions you should be able to building the model and see the results, applied to our friend, the MNIST handwriting dataset.

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt

# Load and preprocess the data
(X_train, y_train), (X_test, y_test) = mnist.load_data()

# TODO 1: Normalize the data to the range [0, 1]
# TODO 2: Normalize the testing data

# TODO 3: One-hot encode the training labels (hint to_categorical)
# TODO 4: One-hot encode the testing labels

# Build the neural network model
model = Sequential([
    Flatten(input_shape=(28, 28)),  # Flatten the input image to a 1D array
    Dense(128, activation='relu'),  # First hidden layer with 128 neurons and ReLU
activation
    Dense(64, activation='relu'),   # Second hidden layer with 64 neurons and ReLU
activation
    Dense(10, activation='softmax') # Output layer with 10 neurons (one for each
class) and softmax activation
])
```

```
# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

# Train the model
history = model.fit(X_train, y_train, epochs=10, batch_size=200,
validation_split=0.2, verbose=2)

# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test, verbose=0)
print(f'Accuracy: {accuracy * 100:.2f}%')

# TODO 5: Make predictions   (hint:  model's predict function)

# Visualize the first 5 test samples and their predicted and actual labels
for i in range(5):
    plt.imshow(X_test[i], cmap='gray')
    plt.title(f'Predicted: {predictions[i].argmax()}, Actual: {y_test[i].argmax()}')
    plt.show()
```

## Deliverables

- Turn in a zip including your **essay** (in PDF format) as well as your source code and screenshots of the program functioning, as follows:
  - An **ipynb** file for Jupyter Notebooks
    - Alternatively, a **py** source file is acceptable as well
  - **Include screenshots of your program working,** placed inside the zip file that you turn in
    - This should include screenshots of the outputs including diagrams and printing of the shapes, evaluation metrics, etc.