Assignment 3: TensorFlow Benchmarking of Architectures and Activations

Due: 02.11.2025

This assignment extends Assignments 1–2. Use the same synthetic house-price dataset unless otherwise stated.

Objective

Design, train, and benchmark multiple TensorFlow/Keras feedforward neural network (FNN) models to study how activation choice, depth (number of layers), and width (neurons per layer) affect training dynamics, accuracy, and runtime.

Dataset

Use the same synthetic dataset as in Assignments 1–2 with at least 500 samples and three input features (e.g., area, number of rooms, location score). Keep the exact same train/validation/test split used previously to enable direct comparison across assignments.

Experimental Design Requirements

You must run a controlled benchmark with the following factors:

- Framework: Use TensorFlow/Keras only (no custom NumPy backprop).
- Activations: Test at least 6 distinct activation functions chosen from: ReLU, LeakyReLU, ELU, SELU, GELU, Swish, Softplus, Tanh, Sigmoid. If you choose LeakyReLU, use it via a layer (tf.keras.layers.LeakyReLU) with alpha=0.01 unless you justify otherwise.
- Depth: Evaluate at least 3 different depths, e.g., 1 hidden layer, 2 hidden layers, 3 hidden layers.
- Width: For each depth, test at least 3 widths per hidden layer, e.g., [8], [16], [32] neurons (you may choose different values).
- Optimization: Use Adam with a fixed learning rate (e.g., 1e-3) for the main grid. You may add a small LR study as a secondary analysis.

Metrics to Record

- Training runtime (seconds) per run.
- Best validation loss (MSE) and corresponding epoch; also report MAE and R^2 on the test set.
- Number of parameters for each model.
- Convergence plots: training vs. validation loss curves for at least the best model of each activation.

Comparison with Assignments 1–2

Using the same dataset split, compare the best TensorFlow models against the earlier one-neuron and the 3-neuron hidden-layer models. Include a small table summarizing results side by side.

Deliverables

- Code: A single runnable notebook (.ipynb) or script (.py) that executes the entire benchmark and saves results to a CSV file.
- Results CSV: One row per run with all metrics and configuration fields (activation, depth, width, runtime, val_MSE, test_MSE, test_MAE, test_R2).
- Figures: (a) Heatmaps or tables summarizing validation MSE across depth×width per activation; (b) Loss curves for the top-3 models; (c) A bar chart comparing test R^2 for the best model of each activation.
- Report (3–5 pages): Brief methods, experimental setup, results (with plots), and discussion including comparisons with Assignments 1–2 and key takeaways.

Grading Rubric (100 points)

- Correctness & Reproducibility (25): Models build/run, logging, and CSV integrity.
- Benchmark Coverage (25): Meets/exceeds the minimal grid (≥6 activations × 3 depths × 3 widths).
- Analysis & Visualizations (25): Clear tables/plots, heatmaps, learning curves; insightful interpretation.
- Comparison with Assignments 1–2 (15): Concrete, data-backed comparison and discussion.
- Code Quality & Report Clarity (10): Clean code, comments, readable report, precise conclusions.

Submission Instructions

Submit a ZIP containing: code, results CSV, figures, and the report (PDF or DOCX). Name your archive as StudentID_Assignment3.zip. Late submissions require prior approval.