Comparison of PSO vs. Gradient Descent in Optimization For Neural Networks

Hunter Price

Motivation & Objectives

1. Particle Swarm Optimization (**PSO**) is very **efficient** when compared with other heuristic based optimization techniques.
2. PSO **does not** require the **gradient** of the problem. This could allow for **non-differentiable activation functions.**
3. PSO is shown to be **extremely parallelizable**.

Methodology

➔ **Datasets:**
- XOR, AND, Synthetic Two-Class Set

➔ **PSO Representation:**
- Network **weights** are flattened and represented as **a particles location**.
- The **Quality function** returns the **loss** of a given batch of data.

➔ **Gradient Descent Counterpart:**
- Uses the **Adam** Optimizer.

➔ **Continuity:**
- Both models used the **same architecture** for a given problem.
- Both models were **hyperparameter tuned** for each problem.

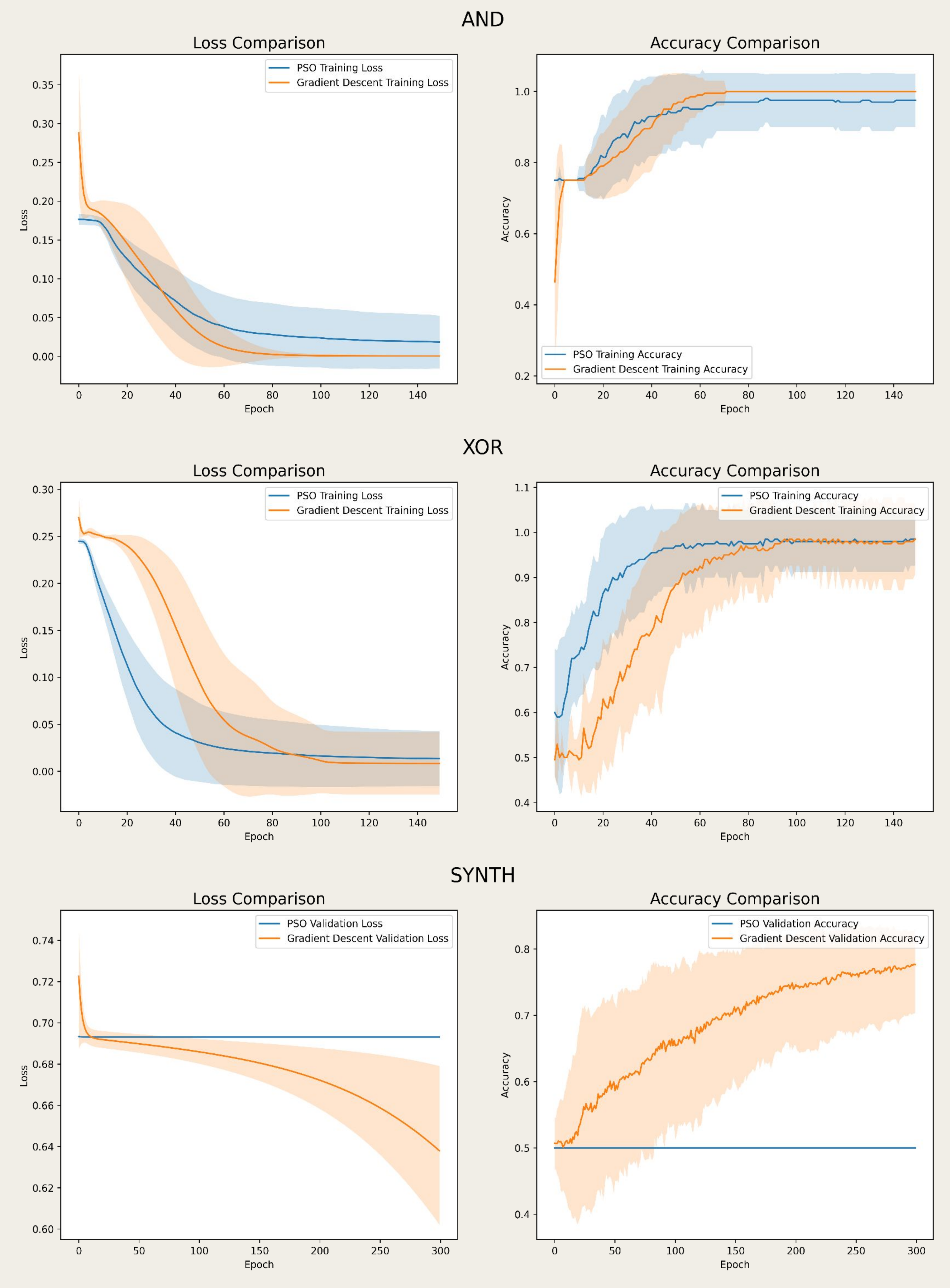
Future Work & Conclusions

1. **PSO** is very **fast** and **effective** on **simple** problems and **outperformed Gradient Descent.**
2. **PSO failed** when any **complexity** was added.
3. Using **Gradient Descent** in **warmup training steps** then **transitioning** to **PSO** could be a avenue to pursue.

Results

➔ **PSO:**
- Finished with **similar** or **better** loss curves for **simple data** such as the **XOR** and **AND** datasets.
- **Failed** to learn on more **complex data** such as the **Synthetic** dataset.

➔ **Gradient Descent:**
- **Dominated** over the **PSO** optimizer when presented with **complex data**.
- **Slower convergence** in **simple problems** than PSO.



THE UNIVERSITY OF TENNESSEE KNOXVILLE