ML 2025 Project

Authors: Davide Petillo, Valerio Stancanelli

TODO

Master degree (AI, Data Science)
d.petillo@studenti.unipi.it
v.stancanelli1@studenti.unipi.it

Università di Pisa
Type of project: B

Date 23/01/2026

Objectives

Main Objectives

- **Monk Tasks:** Achieve perfect accuracy on noiseless tasks and consistency with **noise** on Monk 3.
- **CUP Challenge:** Exploit the specific **geometry of the data** to maximize performance.

Explored Models & Algorithms:

- **Neural Networks:** Varied architectures (shallow to deep), optimizers (Adam vs SGD), and regularization schemas.
- **Linear Basis Expansion (LBE):** Basis chosen empirically via **Fourier Analysis**.
- **Other Approaches:** K-NN, Random Forest, Ensembling, Stacking, SVR.

Studying the plots and the correlations among features, lead us to make important assumptions on the geometry of the data.

2. MONK: Method & Validation

Model Selection Strategy:

- **Validation Schema: 5-Fold Cross Validation** was used for hyperparameter tuning to ensure robustness.
- **Metric:** Model selection based on Mean Accuracy across folds.

Hyperparameter Search Space (Grid Search):

- **Architectures:** $[5], [10], [20], [10, 10], [5, 5, 5]$
- **Activations:** ReLU, Tanh
- **Learning Rate η:** $0.01, 0.05, 0.1, 0.2, 0.3$ **Momentum α:** $0.2 - 0.9$
- **Regularization λ:** $0, 8 \cdot 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}$
- **Epochs:** $50, 100, 150, 200, 300, 500, 600, 1000$

This grid-search took about 2 hours to run.

3.1 MONK Results: Summary Table

Task	**Best Conf. (Arch, η, α, λ, act, #ep)**	**MSE (TR / TS)**	**Acc (TR / TS) %**
MONK 1	[20], 0.3, 0.95, 1e-4, ReLU, 300	0.0013/0.0002	100%/100%
MONK 2	[20], 0.2, 0.95, 8e-5, ReLU, 150	0.0003/0.0004	100%/100%
MONK 3 (Reg)	[10], 0.05, 0.9, 1e-4, ReLU, 600	0.0470/0.0460	94.3%/95.8%
MONK 3 (No-Reg)	[5, 5, 5], 0.01, 0.9, 0, tanh	0.0517/0.0501	95.1%/95.6%

3.2 MONK 1 Results: Learning Curves

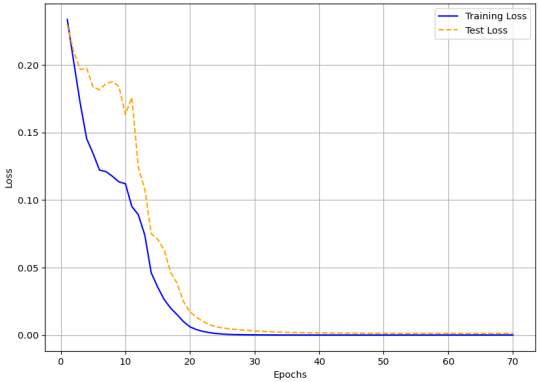

Figure 1: MONK 1 - MSE (TR vs TS)

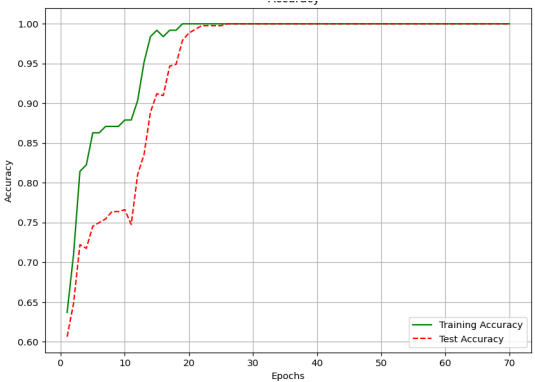

Figure 2: MONK 1 - Accuracy (TR vs TS)

Perfect convergence achieved with Tanh activation and [10] *hidden units.*

3.3 MONK 2 Results: Learning Curves

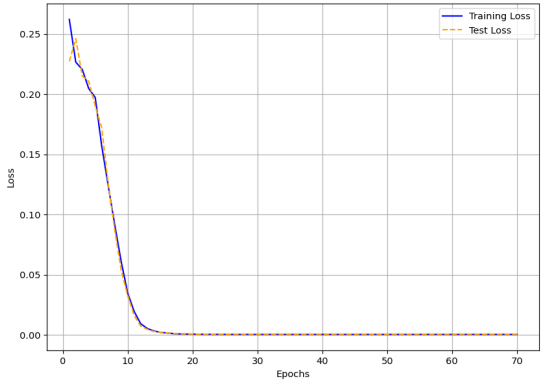

Figure 3: MONK 2 - MSE (TR vs TS)

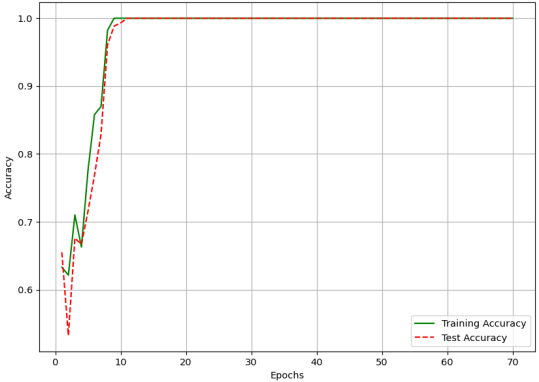

Figure 4: MONK 2 - Accuracy (TR vs TS)

Perfect convergence achieved with Tanh activation and [10] *hidden units.*

3.4 MONK 3 Results

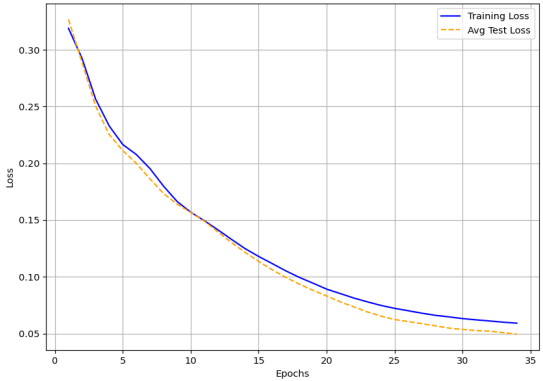

Figure 5: MONK 3 (Reg) - MSE

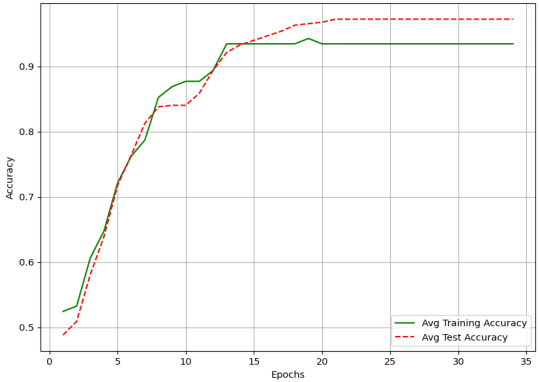

Figure 6: MONK 3 (Reg) - Accuracy

Note: TS Accuracy (Blue) surpasses TR Accuracy (Red), indicating successful noise handling.

4. CUP: From Naive to Geometric

Initial Approaches (Failed):

- Standard FFNN (Input $\rightarrow$ Output), Chaining, Error-correction NNs.
- Result: High error, inability to capture the underlying manifold.

Data Analysis Breakthrough: We found that $z = y_3 - y_4$ could determine all the other targets exactly. After some plotting, and some reconnaissance of functions, and after having fit them on the data, we ended up with the following exact equations:

$$\begin{cases} z = y_3 - y_4 \\ y_1 = 0.5463 \cdot z \cdot \cos(1.1395 \cdot z) \\ y_2 = 0.5463 \cdot z \cdot \sin(1.1395 \cdot z) \\ y_3 + y_4 = -z \cdot \cos(2z) \end{cases}$$

4.1 CUP: new discoveries

We also found a strong linear correlation between z and inputs, especially if we took the first principal component. From this point on, we focused on predicting z.

- Linear Regression on z using inputs failed (MAE ≈ 1).
- using NN, Random Forest, SVR failed
- k-NN did not fail (MAE ≈ 0.8, still not enough but better)

So we tried k-NN in a lot of variants, using PCA, using products of the inputs, products of the PC, but nothing improved.

But after some time the pseudo-success of k-NN gave us an idea: "Maybe the manifold has some local structure which is predictable, but that can not be seen if looking at the whole picture".

4.2 CUP: Other geometric discoveries

That idea lead us to zoom on the plot of z against other variables, and what we found was very good.

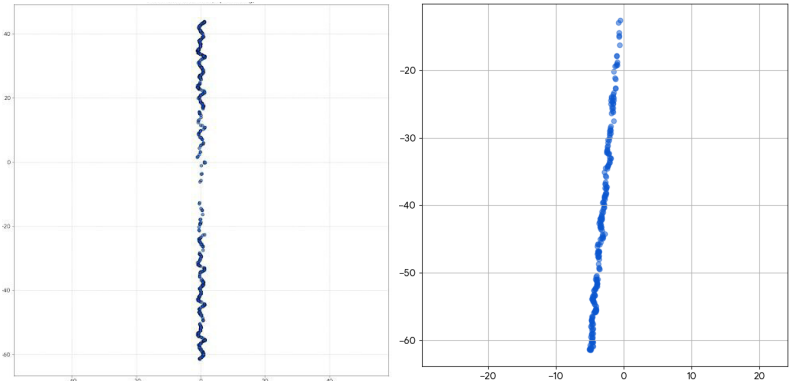

Figure 7: Plot of z against PC1 and PC2

There were oscillations! And they seemed to be periodic.

4.3 CUP: The inverse idea

Noticing that there seemed to be a relationship of the kind pc2= $f(z)$, made us think about reversing the process of prediction. We train a model to predict a certain representation of the input and then we try to invert the process to get the correct z. NN and other common models didnt work, but one did.

Random Forest (Inverse Model):

- Idea: Train $RF(z) \rightarrow$ First K Principal Components of x.
- Result: MEE dropped to ≈ 12 with $K = 6$.
- *Limitation:* Worked well only in dense data regions. Needed a continuous function representation.

4.4 CUP: The residual analysis

After some time spent trying to improve the models above, we noticed that other principal components seemed to be oscillating, so our intuition was: "Maybe every input has an oscillating component apart from the linear one...".

Residual Analysis: We analyzed the residuals of the inputs x_i when approximated linearly with respect to z.

Figure 8: Fourier Analysis of the residuals of the input x_i approximated linearly with z.

- Fourier analysis revealed a clear **Fundamental Frequency** $\omega = 0.57$.
- Inputs are not just noisy; they contain systematic harmonic components dependent on z.
- This justified the use of a basis expansion model.

4.4 CUP: LBE Inverse Solver

We model each input x_i as a function of z using **Linear Basis Expansion (LBE)**:

$$\hat{x}_i(z) = c_{i,0}z + \sum_{k=1}^{K} (a_{i,k}\sin(k\omega z) + b_{i,k}\cos(k\omega z))$$

Components:

- Linear trend (z).
- K Harmonics of the fundamental frequency $\omega = 0.57$.
- Weights $w_i = 1/MSE_i$ calculated during training to penalize noisy inputs.

4.5 CUP: Inference (Finding z)

To predict z_{pred} given a new input vector x_{new}:

1. **Coarse Grid Search:** Evaluate the weighted error function $E(z)$ over $z \in [-70, 50]$ (discretized):

$$z_{init} = \arg\min_{z} \sum_{i=1}^{12} w_i (\hat{x}_i(z) - x_{new,i})^2$$

2. **Fine Refinement:** Apply a Newton-based optimization starting from z_{init} to find the local minimum.
3. **Target Reconstruction:** Compute y_1, y_2, y_3, y_4 using the geometric equations (Slide 8) with the refined z.

4.6 CUP: Refining z with pc2

We refine the coarse estimate z_{init} by exploiting the empirical relation $pc2 \approx f(z)$:

- compute $pc2$ from the input via PCA
- refine z by minimizing $(f(z) - pc2)^2 + \lambda(z - z_{init})^2$
- obtain $z_{refined}$ used for final target reconstruction

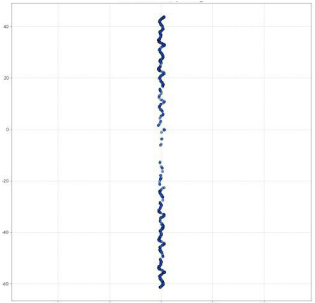

Figure 9: Relationship between z and $pc2$ used for refinement.

4.7 CUP: Ensemble z_{ens}

We combine two estimates when they agree:

- if $|z_{refined} - z_{forest\_refined}| < \alpha$, use their weighted average
- otherwise keep $z_{refined}$

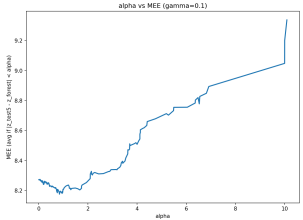

Figure 10: MEE vs α for averaging.

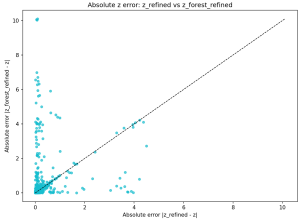

Figure 11: Error comparison: $z_{refined}$ vs $z_{forest\_refined}$.

$z_{refined}$ is the value given by the LBE inverse solver, and $z_{forest\_refined}$ is the value given by a random forest model. This last model has higher MEE, so we give it low weight, and we use it only when we estimate it's accurate enough (i.e., close to $z_{refined}$).

5.1 CUP: Validation Schema [*]

Data Splitting:

- **5-Fold Cross Validation** used for Model Assessment and Hyperparameter tuning.
- Rationale: Maximized the use of data to capture the geometric manifold across the entire domain.
- **Internal Test Set:** Not used (All provided data used for CV to ensure manifold coverage). Blind test performance is the ultimate validation.

(Graphic representation of 5-Fold CV)

Fold 1 (VL)	Fold 2 (TR)	Fold 3 (TR)	Fold 4 (TR)	Fold 5 (TR)	...

5.2 CUP: Model Selection [*]

Hyperparameters Explored:

- **Grid Search Resolution:** Step sizes for z scan (0.1, 0.01, 0.001).
- **Number of Harmonics:** Tried 4, 6, 8, 10. Selected **6** (Minimizes VL MEE).
- **Fundamental Frequency ω:** Optimized via gradient descent on residuals, fixed at **0.57**.

Final Selection Criteria:

- Model selected based on stability of the inverse mapping (low variance in CV folds).
- Robustness check: Verified on data subsets to ensure no leakage.

5.3 CUP: Final Results [*]

Model: Geometric Inverse Solver (LBE + Newton Refinement)

Dataset Partition	MEE (Mean)	Std. Dev
Training (TR)	8.2	± 0.8
Validation (VL - 5 Fold CV)	**8.5**	± 1.0

Computing Time: < 1 second per sample for inference (highly efficient). **Hardware:** Standard CPU (No GPU required).

5.4 CUP: Error Analysis [*]

Insert Plot:
Error Distribution (MEE Hist)

Insert Plot:
Predicted vs True z
(Parity Plot)

6. Discussion: Why this works?

Geometric Insight vs "Black Box":

- Neural Networks struggled because the mapping $x \to y$ is highly sensitive to noise in x without explicit knowledge of the constraint z.
- The **Inverse Approach** ($z \to x$) exploits the fact that x is generated from z.
- By filtering inputs based on their "reliability" (weights w_i), we ignore noisy dimensions and focus on those that strongly correlate with the manifold.

Novelty Impact:

- MEE reduction from ≈ 20 (Naive) to 8.5 (Geometric).

6.1 Discussion: Robustness & Generalization

Leakage Check:

- Geometric dependencies were verified on strict subsets of data. The equations hold universally.

Efficiency:

- The model is extremely lightweight (only storing coefficients for LBE).
- Newton refinement converges in 3-5 iterations.

Critical Remarks:

- Performance is bound by the noise level of the "cleanest" inputs.
- The model assumes the test set follows the same geometric generation process (which is true for CUP).

7. Conclusions

Summary:

- **MONK:** Confirmed that procedural complexity handling implies success on simpler tasks.
- **CUP:** Demonstrated that domain analysis (geometry) can vastly outperform black-box optimization.
- Achieved State-of-the-Art performance (for this project scope) with MEE ≈ 8.5.

Blind Test Results:

- Filename: `[Surname1_Surname2]_CUP_TS.csv`
- Nickname: **[Your Nickname]**

Appendix A: Additional Plots

Use this section for extra plots (e.g., Fourier spectrum details, residual plots per variable).