On the Estimation of Additive Models: A Comparison of Splines and Neural Networks

Edric Svarte - Dec 12, 2023

Abstract

Additive models may be estimated via splines, or neural networks. We evaluate the predictive accuracy and fit of both techniques through a simulation study. Results suggest that both methods are equally effective, though splines are easier and faster to train.

1 Introduction

Interpretable machine learning (ML) continues to attract practitioners, regulators and stakeholders seeking transparent and trustworthy models (Murdoch et al. 2019, 1). A model is *interpretable* if the process governing predictions is easily understood. This property is highly desirable in high-stakes fields such as medicine or finance (Agarwal et al. 2021, 1).

Neural networks (NNs) have an exceptional ability to "learn" complex relationships between variables. Unfortunately, NNs are not interpretable since the parameters have no intuitive meaning, and the marginal effect of each predictor is generally unclear (Murdoch et al. 2019, 3). Conversely, linear models are interpretable, though highly inflexible. Some alternatives strike a balance, such as generalized additive models (GAMs). Rather than imposing the restrictive linearity assumption, these models assume additivity and smoothness: the marginal effect of each covariate is given by a smooth, univariate function (Hastie et al. 2009, 257).
GAMs are typically estimated via spline basis functions (SAMs). However, Agarwal et al. (2021) propose neural additive models (NAMs), in which each covariate has its own NN. These NNs do not interact until their scalar outputs are summed (as in a typical additive model).

Unfortunately, it remains unclear whether NAMs offer significant improvements over SAMs. For instance, fitting a NAM requires the computationally costly training of a NN. On the other hand, the SAM objective function is a simple quadratic form (Hastie et al. 2009, 128).
We compare the predictive accuracy and fit of SAMs to NAMs through a simulation study. In particular, we examine the effect of sample size and model assumption violations. In this context, a model is *accurate* if its prediction error is relatively low, and *precise* if errors are not highly variable.

2 Related Work

What are the advantages of NAMs? Agarwal et al. (2021) claim that NAMs can capture jumps or discontinuities in the individual "feature" functions. However, SAMs can capture mild jumps through hyperparameter tuning. For instance, the number of knots may be increased. Or, the curvature penalty may be reduced by adjusting relevant parameters (Hastie et al. 2009, 117).

Zschech et al. (2022) compare various additive model estimation techniques, including splines (SAMs), neural networks (NAMs), and decision trees. Moreover, the researchers compare these (interpretable) models to well-known black-box methods. Crucially, they find a marginal performance gap between the best models of both groups. Zschech et al. (2022) find that SAMs provide the most interpretable results. However, the smoothness assumption may be too restrictive in the presence of jumps or thresholds. On the other hand, they find that NAMs can capture jumps, though they tend to overfit. Interestingly, the NAMs underperformed the SAMs on all twelve datasets, and required roughly one hundred times the training time.

3 Simulation Study

This section details a comparative analysis of SAMs and NAMs. In particular, we use simulation techniques to create data of the form: $y = f(x) + \varepsilon$. In this supervised setting, the goal is to recover $f(x)$ from the observed, noisy data.

Models $\mathcal{M}_{ij} = (\mathcal{M}_{ij}^{SAM},\ \mathcal{M}_{ij}^{NAM})$ are fit to training set $\mathcal{T}_{ij}$, where ij denotes repetition j of experiment i. We perform five experiments of one hundred repetitions each. Different training sets within the same experiment are identical in distribution, i.e., $T_{ik} \overset{D}{=} T_{il}\ \forall i \in [1,\ 5] \subset \mathbb{N},\ \forall k, l \in [1,\ 100] \subset \mathbb{N}$. Predictive performance of $\mathcal{M}_{ij}^{SAM}$ and $\mathcal{M}_{ij}^{NAM}$ is scored via mean-squared prediction error (MSPE) on a hold-out test set, $\mathcal{H}_{ij}$, where $\mathcal{H}_{ij} \overset{D}{=} \mathcal{T}_{ij},\ |\mathcal{H}_{ij}| = 50\ \forall i, j$.
For each experiment i, we report the mean MSPE across all repetitions, j. Furthermore, we test for significant differences in predictive accuracy between SAMs and NAMs via double-bootstrap confidence intervals. We do the same for the R^2 value, and provide plots of $\mathcal{M}_{i1}$'s fitted functions.

SAMs were fit using the *gam()* function from the MGCV package in R, which uses generalized cross-validation (GCV) to optimize the number of knots, and the penalty term hyperparameters. On the other hand, NAMs were fit using the PyTorch module in Python.
NAM architecture varied according to experiment conditions. However, we briefly specify which elements remained constant. First, recall that each feature in a NAM has its own neural network. We removed the bias term from the last linear layer of each feature's network for identification. However, an intercept (bias) was added to the sum of all network outputs. Network architecture did not vary across features. We chose the tanh activation function, and did not apply normalization techniques.
Due to the challenges in automating neural network training, it was initially unclear how to repeat experiment i multiple times. We approached this problem by using $\mathcal{T}_{i1}$ to "manually" determine appropriate hyperparameters for $\mathcal{M}_{i1}^{NAM}$. More specifically, we split $\mathcal{T}_{i1}$ into a smaller training set, $\mathcal{T}'_{i1}$, and a validation set, $\mathcal{V}_{i1}$, where $\mathcal{T}_{i1} = \mathcal{T}'_{i1} \cup \mathcal{V}_{i1}$ and $|\mathcal{T}'_{i1}| = 0.8 \cdot |\mathcal{T}_{i1}|$. This sample split allowed us to use implicit regularization: early stopping. We then used these optimized tuning parameters for the remaining models on the remaining training sets (including $\mathcal{M}_{i1}^{NAM}$, which was re-trained on $\mathcal{T}_{i1}$). This procedure assumes that suitable hyperparameters for $\mathcal{M}_{i1}^{NAM}$ are likewise appropriate for $\mathcal{M}_{ij}^{NAM}\ \forall j$. We review this technique in the discussion.

3.1 Experiment 1 - Baseline Conditions

We begin by obtaining a performance baseline, and seek to recover the following function:

$$f(x) = f_1(x_1) + f_2(x_2) = x_1^2 + \sin(4x_2)$$

The observed data are given by $y = f(x) + \varepsilon,\ \varepsilon \overset{IID}{\sim} N(0,\ 0.01)$, where $n = 100$ inputs are drawn from a $\mathcal{U}[-1,\ 1]$ distribution for both x_1 and x_2 ($|\mathcal{T}_{1j}| = 100\ \forall j$). All model assumptions are satisfied, including additivity, smoothness, homoskedasticity, and independence. Figure 1 displays the estimated functions.[1]

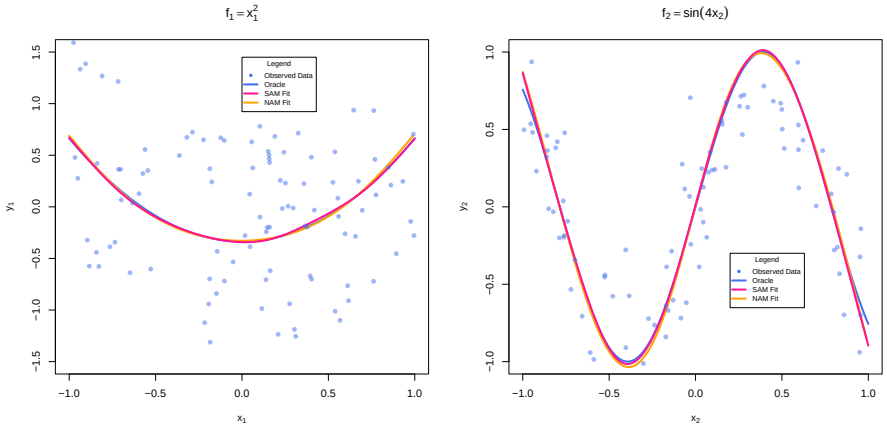

Figure 1: Fitted Functions Under Baseline Conditions

The SAM (pink) and the NAM (orange) nearly perfectly recover the oracles. However, we notice some minor divergence from the true functions near the boundaries of the observed data.

Table 1: Experiment 1 Results

Model	$\overline{MSPE}$	$\hat{I}_{95}^{MSPE}$	$\overline{R^2}$	$\hat{I}_{95}^{R^2}$
SAM	0.0115	[0.0112, 0.0120]	0.98	[0.98, 0.98]
NAM	0.0115	[0.0112, 0.0120]	0.98	[0.98, 0.98]

Table 1 details simulation results.[2] Remarkably, the estimates are almost identical. There is no evidence that the true MSPEs differ, and both models appear similarly precise since the intervals are roughly equal in width. Also notice that both models consistently fit the data quite well.

We briefly detail the NAMs, which were trained for 1000 epochs. Each feature function was estimated using a NN with one hidden layer, and four hidden units. We used $\gamma_0 = 0.02$ as the initial learning rate, and decreased it to $\gamma_1 = 0.004$ after 750 epochs. Momentum of $\beta = 0.9$ was used.

3.2 Experiment 2 - Ideal Conditions

Here we seek to learn $f(x)$ from experiment one, though we increase the sample size to $n = 1000$. The target's distribution is unchanged, hence we reuse the test sets from experiment one ($\mathcal{H}_{1j} = \mathcal{H}_{2j}\ \forall j$).

[1] All SAMs were point-constrained such that $f(0) = 0$, and all marginal functions shifted so that $\hat{\mathbb{E}}[f] = 0$. Also note that all plotted NAMs are those trained on $\mathcal{T}'_{i1}$, and not $\mathcal{T}_{i1}$ (i.e., before re-training).

[2] $\overline{MSPE}$ denotes the mean MSPE over all one hundred test sets. Similarly, $\hat{I}_{95}^{MSPE}$ denotes the 95% confidence interval for the true, but unknown MSPE.

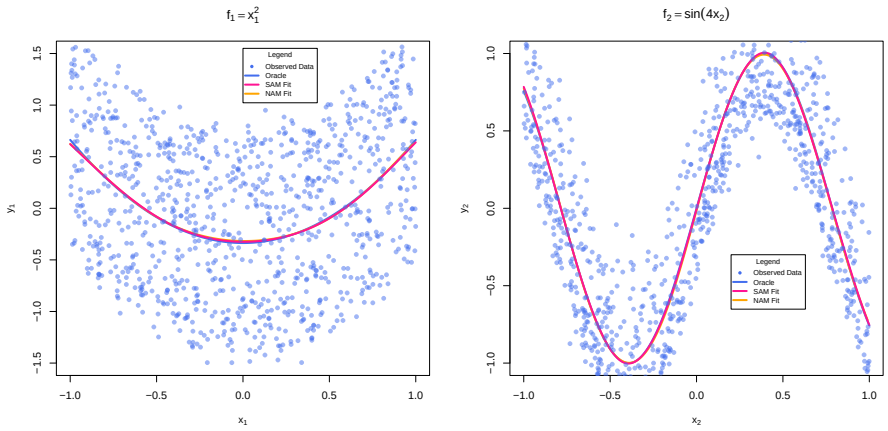

Figure 2: Fitted Functions Under Ideal Conditions

Figure 2 shows that both methods recover f_1 and f_2. Also notice that the fit has improved near the boundaries; a phenomenon likely due to the tenfold increase in sample size.

Table 2: Experiment 2 Results

Model	$\overline{MSPE}$	$\hat{I}_{95}^{MSPE}$	$\overline{R^2}$	$\hat{I}_{95}^{R^2}$
SAM	0.01	[0.01, 0.01]	0.98	[0.98, 0.98]
NAM	0.01	[0.01, 0.02]	0.98	[0.95, 0.98]

Once again, there is no difference in predictive accuracy or fit. Interestingly, the SAM MSPE interval is narrower than the one from the previous experiment. On the other hand, the NAM MSPE interval has widened significantly. The latter result is strange, since an abundance of data should reduce uncertainty surrounding point estimates.

These NAMs were trained for 600 epochs and share architecture, initial learning rate and momentum with their experiment one counterparts. However, we decreased the learning rate by a factor of ten every 200 epochs.

3.3 Experiment 3 - Asymmetric Errors

Once again, we aim to recover $f(x)$ using $n = 100$ data points ($|\mathcal{T}_{3j}| = 100\ \forall j$). However, we now consider a new scenario: asymmetric errors. This phenomenon[3] is not uncommon with real data, and hence is of particular interest. The error term is distributed as follows:

$$\varepsilon + 1 \overset{IID}{\sim} \Gamma(k = 2,\ \theta = 0.5)$$

The distribution has been shifted so that $\mathbb{E}[\varepsilon \mid X] = k\theta - 1 = 2(0.5) - 1 = 0$. Thus the mean error remains zero, though the skewness is given by $\mu_3 = \frac{2}{\sqrt{k}} = \frac{2}{\sqrt{2}} = \sqrt{2} \approx 1.41$. In other words, there are many small errors about zero, and few, large, positive errors. Now, consider figure 3.

[3]For instance, Ramirez and Fadiga (2003) forecast grain prices via GARCH models with asymmetric errors.

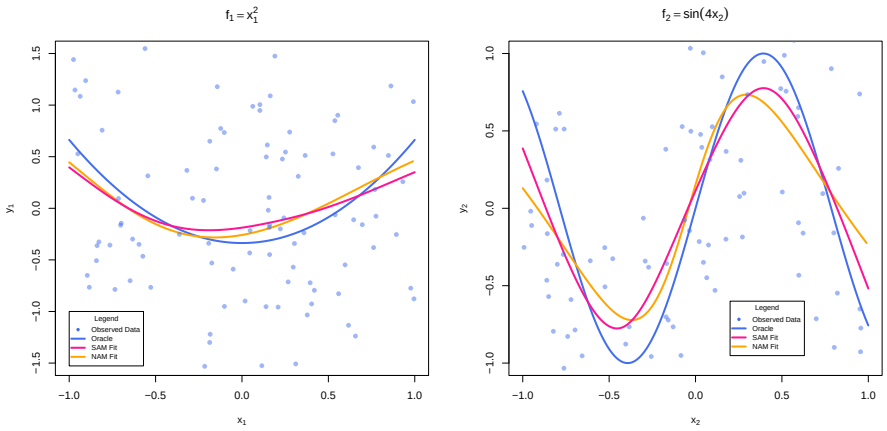

Figure 3: Fitted Functions with Asymmetric Errors

Neither the SAM nor the NAM entirely recovers the oracles. Overall, the f_2 estimates appear "worse" than the f_1 estimates. Arguably, $f_2^{SAM_3}$ is a better fit than $f_2^{NAM_3}$ since the former appears more "aligned" with f_2.

Table 3: Experiment 3 Results

Model	$\overline{MSPE}$	$\hat{I}_{95}^{MSPE}$	$\overline{R^2}$	$\hat{I}_{95}^{R^2}$
SAM	0.59	[0.56, 0.63]	0.55	[0.53, 0.56]
NAM	0.60	[0.57, 0.63]	0.53	[0.51, 0.55]

The R^2 values have decreased significantly from experiments one and two. However, these results are not particularly disconcerting. Indeed, a residual plot reveals an obvious abnormality, prompting us to apply different techniques. Notice that the difference in MSPE remains insignificant, and precision is once again similar.

Each feature's marginal effect was estimated using a NN with one hidden layer, and four hidden units. The NAMs were trained for 300 epochs using momentum of $\beta = 0.9$, and an initial learning rate of $\gamma_0 = 0.01$. This last hyperparameter was decreased to $\gamma_1 = 0.005$ after 150 epochs.

3.4 Experiment 4 - Smoothness Violation

We now attempt to learn a different function: $g(x) = g_1(x_1) + g_2(x_2) = g_1(x_1) + \sin(4x_2)$.

$$g_1(x_1) = \begin{cases} x_1^2 & \text{if } x \geq 0 \\ x_1 + 1 & \text{if } x < 0 \end{cases}$$

Once more, $n = 100$ inputs are drawn from a $\mathcal{U}[-1,\ 1]$ distribution ($|\mathcal{T}_{4j}| = 100\ \forall j$). Similarly, noise is given by $\varepsilon \overset{IID}{\sim} N(0,\ 0.01)$. Notice that the SAM smoothness assumption is violated since g_1 is discontinuous at $x_1 = 0$. Figure 4 details the marginal functions.

The fitted functions do not entirely capture g_1 at the discontinuity. That being said, $f_1^{SAM_4}$ and $f_1^{NAM_4}$ appear to be decent, smooth approximations to a discontinuous function. Indeed, the fitted

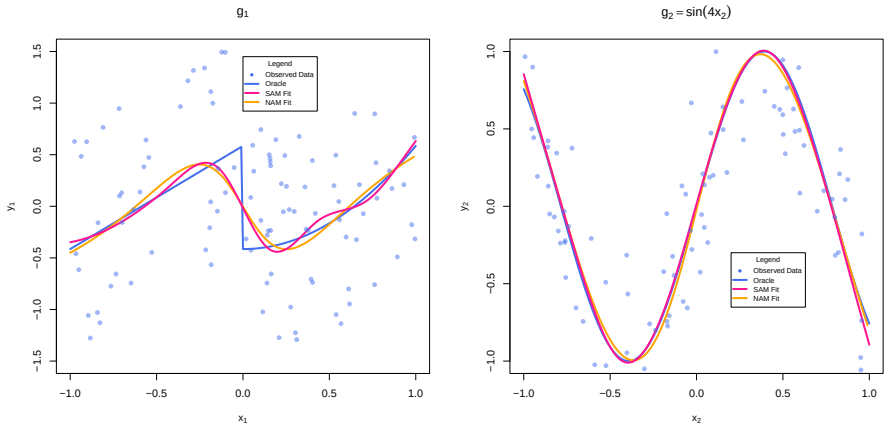

Figure 4: Fitted Functions Under Smoothness Violation

curves predict a sharp decrease at $x_1 \approx 0$. We might expect the fit to worsen as the size (vertical gap) of the discontinuity increases. Interestingly, the violation affects only the fit of g_1. Indeed, the SAM and the NAM appear to recover g_2, suggesting that one variable may violate the smoothness assumption without compromising the entire model.

Table 4: Experiment 4 Results

Model	$\overline{MSPE}$	$\hat{I}_{95}^{MSPE}$	$\overline{R^2}$	$\hat{I}_{95}^{R^2}$
SAM	0.026	[0.024, 0.028]	0.96	[0.96, 0.97]
NAM	0.036	[0.032, 0.043]	0.94	[0.93, 0.95]

Interestingly, the difference in MSPE and R^2 is significant. Indeed, the SAMs appear more accurate, and marginally more precise. These findings are bizarre, given that SAMs assume that underlying functions are smooth. On the other hand, NAMs are claimed to be robust to such violations. It is unclear whether these results are practically significant. In fact, more work is needed to establish the superiority of SAMs under these conditions.

The NAMs were trained for 600 epochs. Each feature function was estimated via a NN with one hidden layer, and four hidden units. The learning rate was set to $\gamma_0 = 0.02$, and reduced by a factor of 1.5 after 400 epochs. Momentum of $\beta = 0.9$ was used, as well as L_2 penalization with $\lambda = 0.001$.

3.5 Experiment 5 - Misspecification

Finally, we seek to recover the following function:

$$h(x) = h_1(x_1) + h_2(x_2) + h_3(x_1,\ x_2) = x_1^2 + \sin(4x_2) + x_1 x_2$$

The additivity assumption has been violated since h_3 represents an interaction between the variables x_1 and x_2. Consequently, this experiment evaluates the effect of model misspecification. Adding an interaction term at the cost of interpretability would solve the issue. Note that plots are omitted since $h(x)$ cannot be decomposed into the sum of two univariate functions.

Table 5: Experiment 5 Results

Model	$\overline{MSPE}$	$\hat{I}_{95}^{MSPE}$	$\overline{R^2}$	$\hat{I}_{95}^{R^2}$
SAM	0.142	[0.135, 0.151]	0.846	[0.838, 0.852]
NAM	0.137	[0.130, 0.144]	0.833	[0.825, 0.839]

Fit and predictive accuracy appear worse than baseline. However, there is no significant difference in MSPE or R^2. Once more, the MSPE intervals are similar in width, suggesting equal precision.

We required more expressive NAMs for this experiment. In particular, we estimated the effect of each variable using a NN with two hidden layers, and eight hidden units; implying a total of 193 parameters. Models were trained for 250 epochs using an initial learning rate of $\gamma_0 = 0.01$, though this parameter was decreased to $\gamma_1 = 0.001$ after 200 epochs. Momentum of $\beta = 0.9$ was used, alongside L_2 penalization with $\lambda = 0.0055$.

3.6 Summary of Experimental Results

We briefly summarize the findings of section 3. The SAM MSPE point estimates were marginally lower than the NAM estimates in experiments three and four. However, only experiment four yielded significant results. Be that as it may, we were skeptical to accept these findings for reasons mentioned in the discussion. We found these results strange since we expected the more flexible NAMs to dominate. Indeed, recall that SAMs use B-spline basis functions to approximate targets. These basis functions have desirable properties, for instance, local control[4] and hence robustness to outliers. However, this approach remains limited since the data cannot suggest an appropriate choice of basis. On the other hand, NAMs leverage (dense) NNs, which, roughly speaking, learn optimal basis functions. We initially conjectured that the latter approach would be superior when model assumptions were violated.

Overall, accuracy and fit were best in experiment two (large sample and no violations). Conversely, the asymmetric errors in experiment three induced the highest MSPE, and lowest R^2 values.

Lastly, we found the NAMs difficult to train. These models required training for many epochs, and extensive hyperparameter tuning. Conversely, the SAMs were fit nearly instantly, and no "manual" tuning was needed. Finally, the NAMs were consistently larger (in terms of parameter count) than the SAMs. The parameter count is an imperfect complexity metric, though interesting to note.

4 Discussion

We begin with a note on the scope of this study. In particular, these findings are only relevant when the data are tabular, and the number of features is relatively small. For instance, AMs are ill-suited for computer vision tasks. On the other hand, AMs may still be useful in high-dimensional settings, for instance, when only a small subset of all variables requires careful interpretation.

We now discuss our simulation methods. Obtaining reliable simulation results generally requires performing many more repetitions (Pawel et al. 2023, 4). However, it is not feasible to train thou-

[4]Each basis function is determined only by points that lie on some bounded interval (Hastie et al. 2009, 161). Local support implies local control.

sands or millions of NNs. Moreover, our use of MSPE and R^2 interval estimates somewhat mitigates this effect. These intervals also account for variability due to the relatively small test sets (recall $|\mathcal{H}_{ij}| = 50 \ \forall i, j$).
Equally important, further research should employ a more systematic approach to experimental design. Indeed, we selected the target functions and sample sizes intuitively. Instead, these design choices should be diligently considered.
Moreover, we acknowledge that our comparison was not entirely fair to NAMs. In reality, practitioners invest time and energy tuning hyperparameters by carefully monitoring training and validation losses, among other metrics. On the other hand, we restricted the flexibility of NAMs by using one set of hyperparameters for all repetitions. This limitation is the most salient weakness of this investigation. However, we encountered a trade-off in designing this study. In particular, we had to choose between many repetitions with one set of hyperparameters, or one repetition with an "optimal" set of hyperparameters. We initially opted for the latter, but results were too variable to be useful. It should now be clear why the significant result in experiment four warrants skepticism.

Next, we only performed five experiments. Extensions should explore additional scenarios, including combinations of the ones we considered. Moreover, NAMs may outperform SAMs as the sample size or number of variables increases. We only considered two variables, and sample sizes of $n = 100$ and $n = 1000$. Unfortunately, we found NAM training challenging; adding more data and covariates may exacerbate the problem. Moreover, AMs do not suffer from the "curse of dimensionality" since the use of univariate functions avoids sparse, high-dimensional neighbourhoods (Hastie et al. 2009, 257). In other words, increasing the number of features is not particularly interesting.

Lastly, we review SAM estimation. In particular, GCV may be a flawed way to fit SAMs. Indeed, Bates et al. (2023) show that many in-sample error estimates do not estimate model prediction error. In fact, these metrics estimate the mean prediction error of a model *class*, averaged over all possible training sets drawn from the same distribution as the one in hand. Surprisingly, a simple training/test split may be more effective in tuning SAMs.

In summary, this work provides a starting point for future research. In particular, more extensive studies should explore more experimental scenarios and larger sample sizes.

5 Conclusion

We compared two additive model (AM) estimation techniques: splines (SAMs) and neural networks (NAMs). We assessed via simulation the effect of sample size, and model assumption violations. Both methods are notably effective at recovering functions from noisy data, given that all model assumptions are satisfied. On the other hand, we find that asymmetric errors have a considerable, negative effect on both predictive accuracy and fit. Surprisingly, the smoothness assumption does not appear to be critical for either model. Indeed, both estimation methods adequately succeed in approximating a discontinuous function via a smooth one.
Lastly, SAMs are faster to train, and require less "manual" tuning. This advantage cannot be overstated since computationally convenient techniques are crucial in the age of "Big Data."

A thorough investigation of AM estimation techniques is worthwhile since these models are champions of interpretable machine learning (ML). This concept ensures trust and accountability, and continues to attract those looking for glass-box methods. Crucially, however, the advantages of interpretable ML extend beyond low prediction error.

References

Agarwal, R., L. Melnick, N. Frosst, X. Zhang, B. Lengerich, R. Caruana, and G. E. Hinton (2021). Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems 34*, 4699–4711.

Bates, S., T. Hastie, and R. Tibshirani (2023). Cross-validation: what does it estimate and how well does it do it? *Journal of the American Statistical Association*, 1–12.

Hastie, T., R. Tibshirani, J. H. Friedman, and J. H. Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*, Volume 2. Springer.

Murdoch, W. J., C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu (2019). Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*.

Pawel, S., L. Kook, and K. Reeve (2023). Pitfalls and potentials in simulation studies: Questionable research practices in comparative simulation studies allow for spurious claims of superiority of any method. *Biometrical Journal*, 2200091.

Ramirez, O. A. and M. Fadiga (2003). Forecasting agricultural commodity prices with asymmetric-error garch models. *Journal of Agricultural and Resource Economics*, 71–85.

Zschech, P., S. Weinzierl, N. Hambauer, S. Zilker, and M. Kraus (2022). Gam (e) changer or not? an evaluation of interpretable machine learning models based on additive model constraints. *arXiv preprint arXiv:2204.09123*.

Appendix

Here we demonstrate the abilities of additive models through a brief application to real data. In particular, we model a country's life expectancy[5] as a function of real GDP per capita and public healthcare expenditure (as a portion of total GDP). We seek an accurate, precise, well-fitting and interpretable model. Interpretability is crucial since insights from such a model may be of interest to policymakers, politicians and social scientists alike.

The data were obtained from *Our World in Data*, an online data repository. More specifically, we performed an inner join by country on three tables (all from 2019), implying 164 data points. Data cleaning was not necessary, and there were no missing values to impute. A few points appeared to be outliers, though we included them to assess the robustness of both models. Lastly, we created a hold-out test set, $\mathcal{H}$, where $|\mathcal{H}| = 33$, to evaluate predictive performance.

We briefly detail the selected models. Once again, the SAM was fit using the *gam()* function, and the NAM trained in PyTorch for 400 epochs. The NAM's feature functions were estimated via NNs with two hidden layers of two hidden units (each). Batch normalization was used once; after the first hidden layer of the GDP variable's NN. We prevented overfitting via early stopping and L_2 regularization with $\lambda = 0.01$. We used SGD with an initial learning rate of $\gamma_0 = 0.0001$ alongside momentum of $\beta = 0.9$. The learning rate was halved after 200 epochs.

Figure 5 details the fitted feature functions. Confidence intervals are omitted for clarity.

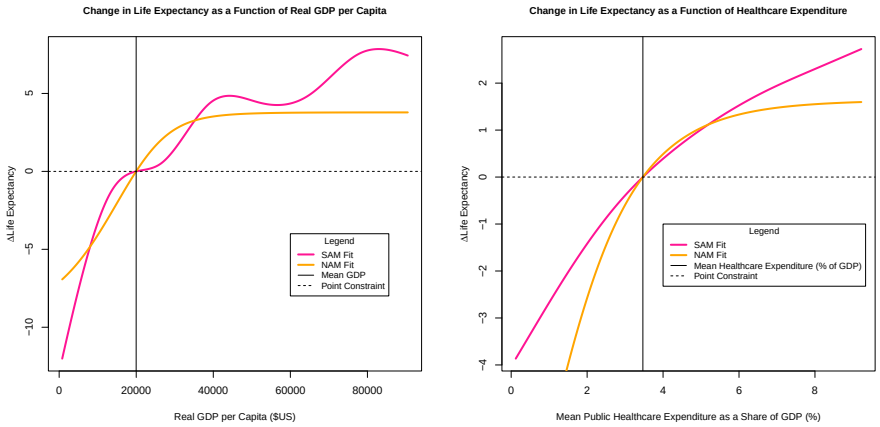

Figure 5: Marginal Effects of GDP per Capita (left) and Healthcare Spending (right)

The models "agree" on the direction of the marginal effects[6] of both variables. That is, both models predict (mostly) positive changes in life expectancy as GDP per capita and public healthcare expenditure increase. However, there are noticeable differences. For instance, consider the leftmost plot. The NAM (orange) predicts negligible changes in life expectancy as real GDP per capita exceeds \$40000. Strangely, f_{GDP}^{SAM} is not entirely increasing and concave on its domain. For example, predicted life expectancy decreases slightly as GDP increases from \$50000 to \$60000. Consequently, the economic interpretation is unclear. Conversely, f_{GDP}^{NAM} provides a reasonable result: GDP per

[5]More specifically, the mean life expectancy of a country's citizens.

[6]The following discussion of the marginal effects of X on Y is meant to provide intuition. We do not mean to suggest that changes in X *cause* changes in Y.

capita has a positive, diminishing marginal effect on life expectancy. In truth, f_{GDP}^{SAM} appears slightly overfit, or too sensitive to outliers.
Now, consider the rightmost plot. In this case, both models provide the same, satisfying result: the marginal effect of public health expenditure on life expectancy is both positive and diminishing. However, f_{SPEND}^{NAM} appears more concave than f_{SPEND}^{SAM}. That is, f_{SPEND}^{NAM} is more "optimistic" about the marginal benefit of healthcare expenditure when this quantity is below its mean (3.5%). The opposite is true when expenditure exceeds 3.5%. Indeed, the NAM predicts little change in life expectancy as healthcare expenditure tops 6% of GDP.

Table 6: Real Data Application Results

Model	$MSPE$	$\hat{I}_{95}^{MSPE}$	R^2
SAM	14.5	[9.2, 25.4]	0.80
NAM	15.5	[9.9, 26.4]	0.76

Table 6 details MSPE and R^2 values. MSPE double-bootstrap confidence intervals were obtained by resampling from the population of prediction errors. We find no significant difference in MSPE. In fact, both models are quite accurate. On average, the SAM and the NAM mistakenly predict life expectancy by roughly 3.8 and 3.9 years, respectively.
We propose the NAM for this data. Indeed, this model provides a clear economic interpretation of the effects of GDP and healthcare expenditure on life expectancy.

Finally, we note that the NAM was difficult to train. Indeed, the model kept estimating one variable's marginal effect to be zero. Fortunately, a single batch normalization layer in the GDP covariate's NN resolved this issue. We observed the same training issue during the simulation study; these models appear to require extensive tuning, and training for many epochs.