Learning Activation Functions in Neural Networks

Gated Mixtures, Temperature/Entropy Control, and Stability-Aware Evaluation

Seher Emir
Repository: `https://github.com/seheremir/Learning-Activation-Functions-in-Neural-Networ`

December 2025

Abstract

Activation functions are commonly treated as a fixed design choice (e.g., ReLU, Tanh, Sigmoid) despite their strong influence on optimization dynamics and representational capacity. This project approaches activation behavior as a learnable component by introducing gated mixtures of candidate nonlinearities, complemented with temperature scaling and entropy regularization to control selectivity and diversity. Using a controlled MLP backbone and a consistent training protocol, we compare fixed-activation baselines to multiple learnable activation variants, report Accuracy, F1, and ROC-AUC, and emphasize robustness by aggregating results across multiple random seeds. In addition to performance, we present interpretability analyses that reveal what the gating mechanism learns at the layer and sample level, and we provide a fully reproducible pipeline with export-ready plots and tables.

1 Introduction

Activation functions provide the nonlinearity that enables neural networks to approximate complex mappings. In practice, however, activation choice is usually decided manually and then fixed for all layers and all samples. This is a pragmatic convention: it works, it is computationally efficient, and it is well understood. At the same time, it implicitly assumes that a single nonlinearity is equally suitable across the network, across the dataset, and across different stages of training.

This project investigates an alternative viewpoint: activation behavior can be treated as a learnable component rather than a static hyperparameter. The central idea is straightforward—replace the single activation with a mixture of candidate activations, and let the model learn how to combine them. This can be done at different granularities: globally per layer (static gating) or adaptively per sample (input-dependent gating). From an engineering perspective, this approach is attractive because it remains end-to-end differentiable, can be implemented as a drop-in replacement for standard activations, and provides interpretability through learned mixture weights.

Beyond raw performance, the project emphasizes evaluation practices that are often underreported in small-scale neural network studies. In particular, we explicitly measure stability across random seeds and report aggregated statistics. This allows the conclusions to reflect not only a single run but also typical behavior under reasonable randomness in initialization and training.

2 Method: Learning Activation Behavior

Fixed activation baselines

We first establish reference models that use a single, fixed activation function throughout the hidden layers. Separate baseline runs are performed with ReLU, Tanh, and Sigmoid. These baselines serve two purposes: they provide a clear performance benchmark, and they reveal optimization and calibration characteristics that can be compared against learnable alternatives.

Gated mixtures of candidate activations

Let x denote the pre-activation input to a hidden layer. Instead of applying a single nonlinearity $\phi(\cdot)$, we define a set of candidate activations $\{\phi_k(\cdot)\}_{k=1}^{K}$ and compute a convex mixture:

$$y \;=\; \sum_{k=1}^{K} g_k\,\phi_k(x), \quad \text{with} \quad \sum_{k=1}^{K} g_k = 1,\; g_k \geq 0. \tag{1}$$

The mixture weights g_k are produced by a softmax over logits z_k:

$$g_k \;=\; \mathrm{softmax}\left(\frac{z_k}{T}\right), \tag{2}$$

where $T > 0$ is a temperature parameter. Temperature controls how selective the mixture becomes: smaller values encourage sharp, near one-hot selection, whereas larger values promote smoother combinations. This provides a direct mechanism to explore the trade-off between specialization and blending.

Static vs. input-dependent gating

We consider two gating strategies. In the *static* setting, the logits z_k are learned per layer and shared across all samples, producing a global preference profile for each layer. In the *input-dependent* setting, z_k are generated from the current layer input through a small gating head. This enables sample-specific activation mixtures and can be interpreted as a lightweight mixture-of-experts mechanism applied at the activation level.

Entropy regularization for controlled selectivity

A practical issue in learnable mixtures is collapse: the gate may converge to always selecting one activation, making the mixture effectively identical to a fixed baseline. Collapse is not necessarily wrong, but it reduces the interpretability and may not be desirable if the goal is adaptive behavior. To control this, we introduce entropy regularization over the gating distribution:

$$H(g) = -\sum_{k=1}^{K} g_k \log(g_k). \tag{3}$$

We sweep the regularization strength λ and analyze how it affects mixture diversity, predictive performance, and run-to-run stability. In combination with temperature, entropy provides an explicit dial to tune the model between a selective regime (close to choosing a single best activation) and a diverse regime (blending multiple nonlinearities).

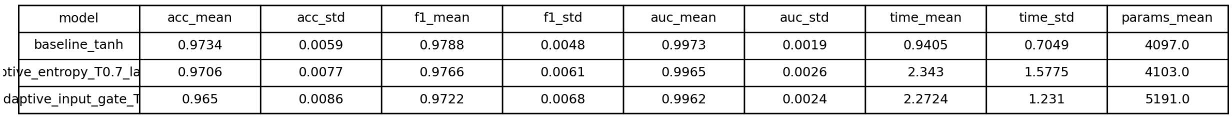
3 Experimental Setup

All experiments are conducted under a consistent pipeline to ensure that comparisons reflect activation strategy rather than incidental differences in training. We use the Breast Cancer dataset for binary classification and standardize features using statistics computed on the training split. The same preprocessing parameters are applied to validation/test data. The backbone architecture is an MLP with fixed layer widths and training hyperparameters across all configurations, and only the activation mechanism is varied.

Evaluation uses multiple complementary metrics. Accuracy provides a direct measure of classification correctness, while F1 score captures the balance between precision and recall. ROC-AUC is included to provide threshold-independent assessment. Because neural networks can exhibit non-trivial variance across random initializations, each configuration is trained across multiple random seeds; results are aggregated using mean and standard deviation, allowing conclusions to rely on typical behavior rather than isolated outcomes.

4 Results and Analysis

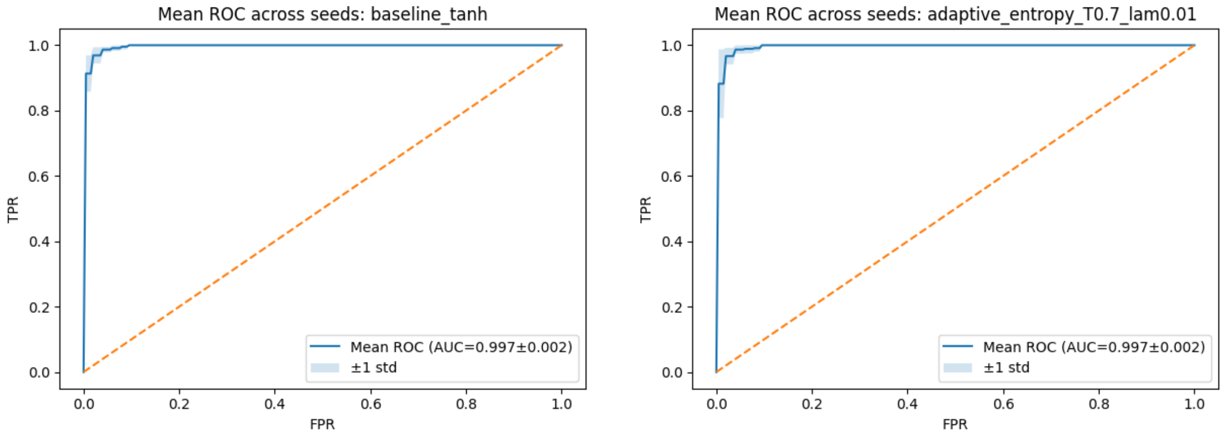
The project produces a structured set of export-ready plots and tables. Figure 1 summarizes overall performance using an aggregated leaderboard across model variants. This figure is designed as a high-level entry point: it enables direct comparison of fixed baselines and learnable strategies under a common evaluation protocol.



model	acc_mean	acc_std	f1_mean	f1_std	auc_mean	auc_std	time_mean	time_std	params_mean
baseline_tanh	0.9734	0.0059	0.9788	0.0048	0.9973	0.0019	0.9405	0.7049	4097.0
ptive_entropy_T0.7_la	0.9706	0.0077	0.9766	0.0061	0.9965	0.0026	2.343	1.5775	4103.0
daptive_input_gate_T	0.965	0.0086	0.9722	0.0068	0.9962	0.0024	2.2724	1.231	5191.0

Figure 1: Aggregated performance leaderboard across fixed and learnable activation strategies.

ROC analysis complements the leaderboard by illustrating separability across decision thresholds. Figure 2 contrasts a strong fixed baseline against a representative learnable activation configuration. The mean ROC curves aggregated over seeds indicate whether improvements are consistent and whether gains appear across the operating range rather than at a single threshold.

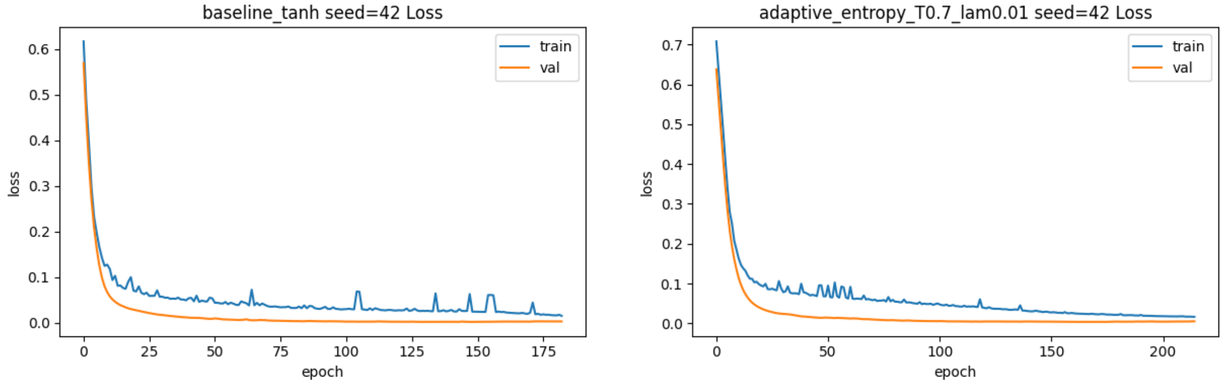


(a) Fixed baseline (Tanh): mean ROC over seeds.

(b) Learnable gating: mean ROC over seeds.

Figure 2: ROC comparison between a fixed activation baseline and a learnable gated mixture configuration.

Training dynamics provide additional evidence about optimization behavior. Figure 3 shows representative loss curves under matched seed settings, illustrating whether the learnable activation mechanism introduces instability, slows convergence, or yields smoother training trajectories compared to fixed activations.

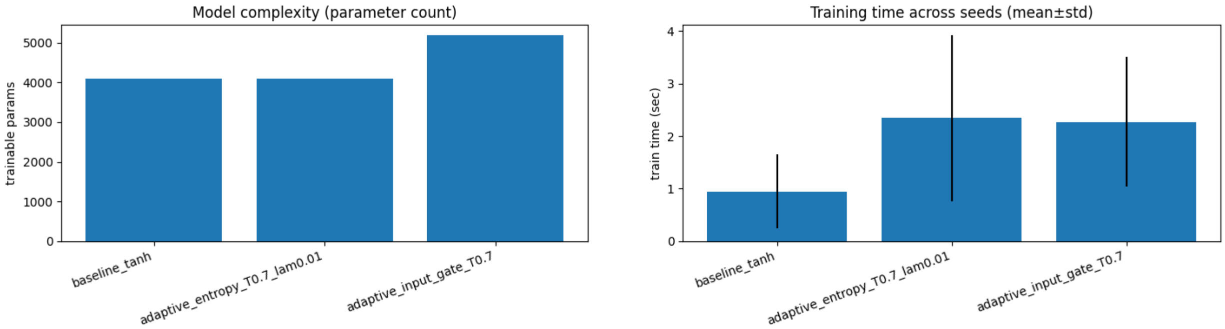


(a) Fixed baseline loss (example seed).

(b) Learnable gating loss (same seed index).

Figure 3: Representative training loss curves for fixed vs. learnable activation strategies.

Any learnable mechanism introduces potential complexity overhead. We therefore include explicit accounting of parameter count and training time. Figure 4 quantifies the cost of moving from fixed activations to learnable mixtures and provides a practical view of the efficiency-performance trade-off.

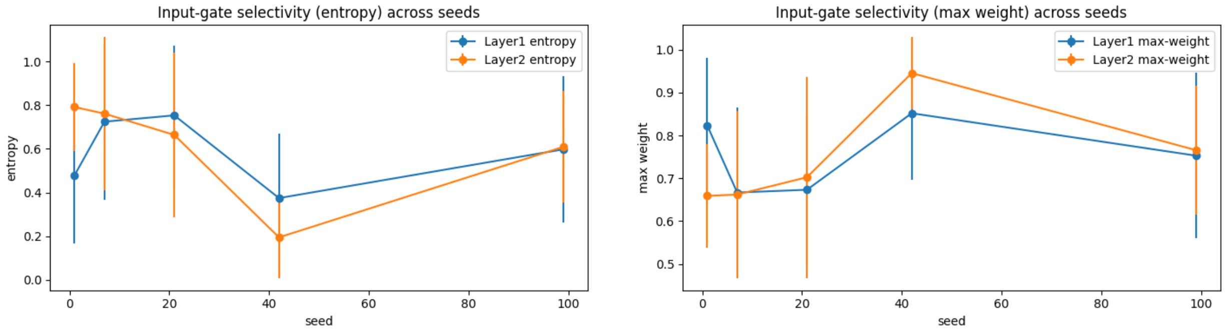


(a) Parameter count by model variant.

(b) Training time (mean ± std) across seeds.

Figure 4: Complexity and runtime trade-offs introduced by learnable activation mechanisms.

A defining benefit of learnable activations is interpretability. Instead of treating activation choice as hidden engineering intuition, we can directly inspect gating behavior. Figure 5 reports gate entropy and selectivity across seeds. Entropy captures how broadly the gate distributes probability mass across candidate activations, while the maximum gate weight captures how often the model behaves like a near-discrete selector. Together, these plots provide evidence about whether the method truly learns a mixture or effectively collapses to a single activation family.



(a) Gate entropy across seeds (diversity indicator).

(b) Max weight across seeds (selectivity indicator).

Figure 5: Gating behavior across multiple runs, supporting stability-aware interpretation.

5 Implementation and Reproducibility

The implementation is organized around a fully reproducible workflow. Data preparation standardizes inputs and exports preprocessing artifacts to ensure consistent inference behavior. Training scripts support both fixed baselines and learnable activation variants under the same MLP architecture and optimization settings. Each run logs training history, exports model checkpoints, and writes metrics in a structured format. To support stability-aware reporting, experiments are repeated across multiple seeds; results are aggregated into summary tables and mean ROC/confusion matrix visualizations.

The repository provides a report-oriented export pack that includes all necessary assets for presentation and documentation. In particular, the plots and tables referenced in this paper are produced automatically by the pipeline so that the LaTeX document can be compiled without manual figure editing. This design reduces the risk of inconsistencies between reported numbers and underlying logs, and it makes the work suitable for public release in a version-controlled environment.

6 Alternative Approaches to Learning Activations

Learning activation behavior can be approached in several ways, each with different trade-offs in expressiveness, cost, and interpretability. A common line of work uses parametric activation families where shape parameters (e.g., slopes, offsets, curvature controls) are learned alongside model weights. This preserves the simplicity of a single activation per unit while providing limited adaptivity. Another direction uses piecewise or spline-based activations, where the nonlinearity is represented by multiple segments whose values or derivatives are learned under smoothness constraints. This provides high flexibility but requires careful regularization to avoid pathological shapes.

More expressive approaches treat activation computation as an expert-selection problem. Maxout-like mechanisms select the maximum over multiple affine components, effectively learning a complex piecewise linear activation. Kernel-based activations represent nonlinearities through kernel expansions and learned coefficients, offering strong approximation power at the cost of additional computation. Neural activation functions go further by defining the activation itself as a small neural module, but they typically require constraints to maintain stable gradients and computational efficiency.

Within this landscape, gated mixtures provide a practical middle ground. They generalize fixed activations in a clean and interpretable way, remain fully differentiable, and allow direct analysis through the learned mixture weights. Temperature and entropy terms further allow explicit control over whether the gate should behave as a selector or as a blend, enabling principled experimentation instead of ad-hoc tuning.

7 Conclusion

This project demonstrates that activation choice can be made learnable without abandoning practical training pipelines. By comparing fixed baselines to gated mixtures under a controlled backbone and robust seed-aggregated evaluation, we show how activation adaptivity can be studied as an explicit modeling component. The method provides not only competitive predictive performance but also interpretability through gating statistics and practical reporting assets suitable for academic presentation and open-source release.

All code, experiments, logs, and exported figures/tables are available in the public repository:

`https://github.com/seheremir/Learning-Activation-Functions-in-Neural-Networks`