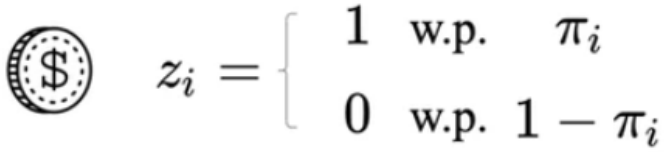
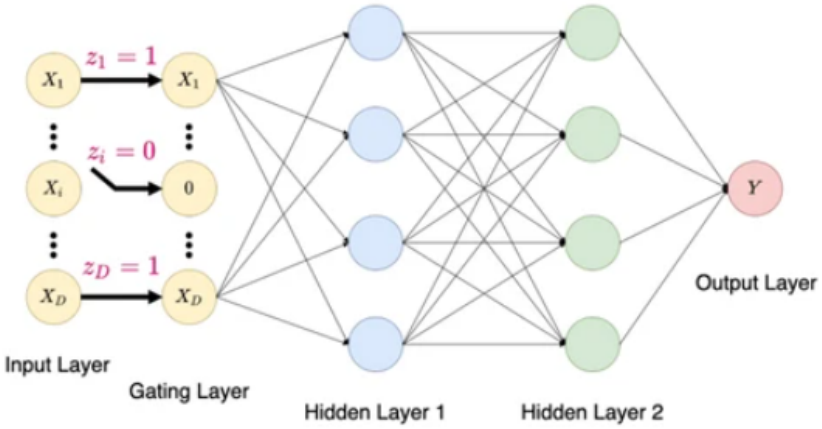
SToG blogpost

October 2025





This post introduces SToG — a compact Python library that brings learnable, stochastic feature selection into end-to-end deep learning. If you face high-dimensional inputs, want interpretable, sparse subsets of features, and still need the expressiveness of nonlinear models, SToG is for you. The library and a technical report with details, intuitions, and training tips are available in the project materials.

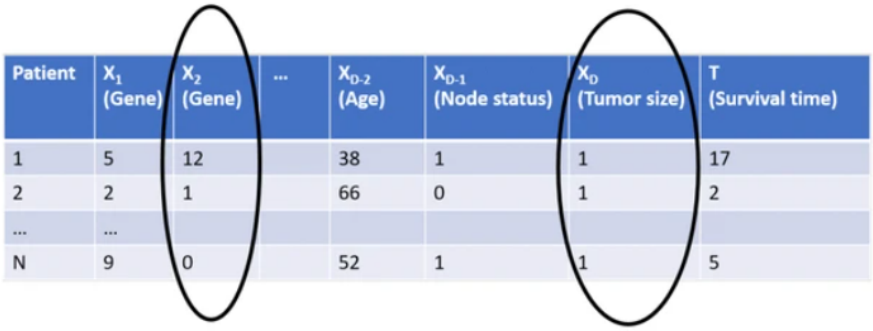
1 Why Feature Selection Matters

Feature selection is an essential step in preparing data for machine learning models. It involves choosing the most relevant features—or variables—that directly influence the output a model tries to predict. Why is this important?

Because real-world datasets often contain many features, but not all contribute meaningful information. Including irrelevant or redundant features can lead to complex models that take longer to train, risk overfitting (fitting noise rather than signal) and are harder to interpret

By selecting a smaller subset of key features, models become simpler, faster, and often more accurate. They are also easier to understand and explain, which is crucial in fields like healthcare or finance where decision transparency matters.

An example of dataset, where one feature is useless.



Patient	X_1 (Gene)	X_2 (Gene)	...	X_{D-2} (Age)	X_{D-1} (Node status)	X_D (Tumor size)	T (Survival time)
1	5	12		38	1	1	17
2	2	1		66	0	1	2
...	...						
N	9	0		52	1	1	5

2 Stochastic Gating for Feature Selection

Stochastic gating is a modern method designed to overcome some of these challenges. It works by associating a stochastic (random) “gate” variable with each feature. Each gate decides probabilistically whether the feature is ”on” (used) or ”off” (ignored) during model training.

Concretely, stochastic gates approximate selecting features by sampling from a relaxed Bernoulli distribution, where the probability of a gate being active is learned continuously. This lets the training process use gradient-based optimization to both learn the model and select features simultaneously.

SToG provides several complementary gating mechanisms you can use independently or mix, depending on your task.

3 StoG library

There will be description of implemented methods, library structure and a small demo.

4 Inspiring examples of application

There will be some examples of application this method and this library for different ML and DL tasks. The structure will be as follows: experimental setup and results, showing that STG is effective for feature selection task, And implementation of this method using our library is user-friendly.