Probabilistic Kolmogorov–Arnold Networks

Andrew Siyuan Chen
Engineering, Cambridge University
sc2178@cantab.ac.uk

May 17, 2024

1 Introduction

Paper [Liu et al. 2024] introduced the idea of using non-linear activation functions to replace traditional linear weight activation for the neurons in a Multi-Layer Perceptron (MLP), creating a Kolmogorov-Arnold Network (KAN). The results are significant, with the model possessing better fitting abilities and being 100 times more parameter efficient.

They used learnable B-Splines as support for the non-linear neurons. Here we extend that idea, replacing the B-Splines with 1-Dimensional Gaussian Processes [Rasmussen and Williams 2006] to create probabilistic non-linear neurons, creating a Probabilistic Kolmogorov-Arnold Network (PKAN).

2 Probabilistic Functions

2.1 Gaussian Process

[Rasmussen and Williams 2006, p.13] defines a Gaussian Process (GP) as:

Definition 2.1. *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

$$\begin{aligned} f &\sim \mathcal{GP}(m, k) \\ \text{where} \quad m(\boldsymbol{x}) &= \mathbb{E}[f(x)] \\ k(\boldsymbol{x}, \boldsymbol{x}') &= \text{Covar}[f(x), f(x')] \end{aligned} \tag{1}$$

The mean of $f(x)$ is defined by mean function $m(x)$, and the covariance of $f(x)$ and $f(x')$ is defined by covariance function $k(x, x')$. Given a vector of known function values $\boldsymbol{h}$ and corresponding input space locations $\boldsymbol{z}$, and a new input location x, the posterior probability distribution of the function output $f(x)$ is:

$$\begin{aligned} p(f(x)|\boldsymbol{h}) &= \mathcal{N}(f(x)|\mu, \Sigma) \\ \text{where} \qquad \mu &= m(x) + \boldsymbol{k}_{xh} K_{hh}^{-1} \boldsymbol{h} \\ \Sigma &= k(x, x) - \boldsymbol{k}_{xh} K_{hh}^{-1} \boldsymbol{k}_{hx} \\ \boldsymbol{k}_{xh}^T = \boldsymbol{k}_{hx} &= \begin{bmatrix} k(x, z_1) \\ k(x, z_2) \\ \vdots \end{bmatrix} \\ K_{hh} &= \begin{bmatrix} k(z_1, z_1) & k(z_1, z_2) & \dots \\ k(z_2, z_1) & & \\ \vdots & & \end{bmatrix} \end{aligned} \tag{2}$$

2.2 Gaussian Process with a Gaussian Input

GP is great for mapping a given input location x to a function output distribution $\tilde{f}_x \sim p(\tilde{f}_x)$. In its basic form however, the input location x is deterministic. If we wish to use GP as non-linear activations in a deep multi-layer PKAN, we need a way to handle $\tilde{x} \sim p(x)$. Below proposes a way of doing so.

First define the mean function to be linear:

$$m(x) = ax + b \tag{3}$$

and the covariance function to be the squared exponential function [Rasmussen and Williams 2006, p.83], which can be rewritten in the form of a Gaussian Distribution:

$$k(x, x') = s^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right) = s^2 l \sqrt{2\pi}\mathcal{N}(x|x', l^2) \tag{4}$$

We also restrict the input distribution to be Gaussian as well, giving:

$$f \sim \mathcal{GP}\left(m(\cdot), k(\cdot, \cdot)\right), \quad \tilde{x} \sim p(x) = \mathcal{N}(x|\mu_x, \sigma_x^2) \tag{5}$$

Defining the output $\tilde{y}$:

$$\tilde{y} = \int f(x)p(x)dx \tag{6}$$

Here $p(x)$ is treated as a weight function that applies a scaling to the GP across the input space. Since each f_x is a Gaussian random variable and integration is a linear operation on f, $\tilde{y}$ will be a Gaussian random variable as well.

Suppose that we have some known data $\boldsymbol{h}$, corresponding to locations $\boldsymbol{z}$, for the GP. Then:

$$\begin{aligned}
\mathbb{E}\left[\tilde{y}|\boldsymbol{h}\right] &= \sqrt{2\pi}s^2 l \boldsymbol{q}_{xh} K_{hh}^{-1}\boldsymbol{h} \\
\mathrm{Var}\left[\tilde{y}|\boldsymbol{h}\right] &= \frac{s^2 l}{\sqrt{l^2 + 2\sigma_x^2}} - 2\pi s^4 l^2 \boldsymbol{q}_{xh} K_{hh}^{-1} \boldsymbol{q}_{hx} \\
\text{where } \boldsymbol{q}_{xh} = \boldsymbol{q}_{hx}^T &= \begin{bmatrix} \mathcal{N}(\mu_x|z_1, \sigma_x^2 + l^2) \\ \mathcal{N}(\mu_x|z_2, \sigma_x^2 + l^2) \\ \vdots \end{bmatrix}^T
\end{aligned} \tag{7}$$

In practice, this corresponds to, for $N \to \infty$

$$y = \mathrm{mean}(\boldsymbol{y}), \quad \boldsymbol{y} \sim p(\boldsymbol{f}|\boldsymbol{x}, \boldsymbol{h}) = \mathcal{N}(\boldsymbol{f}|\mu, \Sigma), \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \quad x_i \sim p(x) \tag{8}$$

Then the mean can be calculated

References

[1] Ziming Liu et al. "KAN: Kolmogorov-Arnold Networks". In: *arXiv:2404.19756 [cs.LG]* (2024). URL: https://arxiv.org/abs/2404.19756.

[2] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. the MIT Press, 2006. ISBN: 026218253X.

Appendix A Proofs & Formulas

A.1 Integrating product of Gaussians

$$\int \mathcal{N}(\boldsymbol{y}|W\boldsymbol{x}+\boldsymbol{b},\Sigma_2)\mathcal{N}(\boldsymbol{x}|\mu,\Sigma_1)d\boldsymbol{x} = \mathcal{N}(\boldsymbol{y}|W\mu+\boldsymbol{b}, W\Sigma_1 W^T + \Sigma_2) \tag{9}$$

Equation 9 can be proven by considering Gaussian random variable $\tilde{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{x}|\mu,\Sigma_1)$ and $\tilde{\boldsymbol{e}} \sim \mathcal{N}(\boldsymbol{e}|\boldsymbol{b},\Sigma_2)$. Then the Gaussian random variable $\tilde{\boldsymbol{y}} = (W\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{e}}) \sim \mathcal{N}(\boldsymbol{y}|W\mu+\boldsymbol{b}, W\Sigma_1 W^T + \Sigma_2)$. Marginalizing the distribution for $\tilde{\boldsymbol{y}}$ gives $p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$, where we note that $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\mu,\Sigma_1)$ and $p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|W\boldsymbol{x}+\boldsymbol{b},\Sigma_2)$, thus proving Equation 9.

A.2 Proof for Equation 7

Given

$$\begin{aligned} m(x) &= ax + b \\ k(x,x') &= s^2 \exp(-\frac{(x-x')^2}{2l^2}) = \sqrt{2\pi}s^2 l\mathcal{N}(x|x',l^2) \\ p(x) &= \mathcal{N}(x|\mu_x,\sigma_x^2) \end{aligned} \tag{10}$$

And defining

$$\boldsymbol{q}_{xh} = \boldsymbol{q}_{hx}^T = \begin{bmatrix} \mathcal{N}(\mu_x|z_1,\sigma_x^2+l^2) \\ \mathcal{N}(\mu_x|z_2,\sigma_x^2+l^2) \\ \vdots \end{bmatrix}^T \tag{11}$$

Applying Equation 9, we have the mean

$$\begin{aligned} \mathbb{E}\left[\tilde{y}|\boldsymbol{h}\right] &= \int p(x)\mathbb{E}\left[f(x)|\boldsymbol{h}\right]dx \\ &= \int p(x)\left(m(x) + \boldsymbol{k}_{xh}K_{hh}^{-1}\boldsymbol{h}\right)dx \\ &= a\int xp(x)dx + b + \begin{bmatrix} \int p(x)k(x,z_1)dx \\ \int p(x)k(x,z_2)dx \\ \vdots \end{bmatrix}^T K_{hh}^{-1}\boldsymbol{h} \\ &= a\mu_x + b + \sqrt{2\pi}s^2 l \begin{bmatrix} \mathcal{N}(\mu_x|z_1,\sigma_x^2+l^2) \\ \mathcal{N}(\mu_x|z_2,\sigma_x^2+l^2) \\ \vdots \end{bmatrix}^T K_{hh}^{-1}\boldsymbol{h} \\ &= a\mu_x + b + \sqrt{2\pi}s^2 l\boldsymbol{q}_{xh}K_{hh}^{-1}\boldsymbol{h} \end{aligned} \tag{12}$$

and the variance

$$\begin{aligned} \mathrm{Var}\left[\tilde{y}|\boldsymbol{h}\right] &= \int\int p(x)\mathrm{Covar}\left[f(x),f(x')|\boldsymbol{h}\right]p(x')dxdx' \\ &= \int\int p(x)k(x,x')p(x')dxdx' - \int\int p(x)\boldsymbol{k}_{xh}K_{hh}^{-1}\boldsymbol{k}_{hx'}p(x')dxdx' \end{aligned} \tag{13}$$

For the first term

$$\begin{aligned} \int\int p(x)k(x,x')p(x')dxdx' &= \int\left(\int p(x)k(x,x')dx\right)p(x')dx \\ &= \int \sqrt{2\pi}s^2 l\mathcal{N}(\mu_x|x',l^2+\sigma_x^2)p(x')dx' \\ &= \sqrt{2\pi}s^2 l\mathcal{N}(\mu_x|\mu_x,l^2+2\sigma_x^2) \\ &= \frac{s^2 l}{\sqrt{l^2+2\sigma_x^2}} \end{aligned} \tag{14}$$

For the second term

$$\begin{aligned}\int\int p(x)\boldsymbol{k}_{xh}K_{hh}^{-1}\boldsymbol{k}_{hx'}p(x')dxdx' &= \int\left(\int p(x)\boldsymbol{k}_{xh}dx\right)K_{hh}^{-1}\boldsymbol{k}_{hx'}p(x')dx' \\ &= \int\sqrt{2\pi}s^2l\boldsymbol{q}_{xh}K_{hh}^{-1}\boldsymbol{k}_{hx'}p(x')dx' \\ &= \sqrt{2\pi}s^2l\boldsymbol{q}_{xh}K_{hh}^{-1}\int\boldsymbol{k}_{hx'}p(x')dx' \\ &= 2\pi s^4l^2\boldsymbol{q}_{xh}K_{hh}^{-1}\boldsymbol{q}_{hx}\end{aligned} \tag{15}$$

Giving the variance in Equation 13 to be:

$$\text{Var}\left[\tilde{y}|\boldsymbol{h}\right] = \frac{s^2l}{\sqrt{l^2+2\sigma_x^2}} - 2\pi s^4l^2\boldsymbol{q}_{xh}K_{hh}^{-1}\boldsymbol{q}_{hx} \tag{16}$$

As a sanity check, we can see that when $\sigma_x^2 = 0$ which indicates that the input x is deterministic, the expressions return to the original GP posterior in Equation 2