روش‌هایی برای حل معادله انتقال یک‌بعدی توسط شبکه‌های عصبی فیزیک‌محور

پژوهشگر: امید خسروی

استاد راهنما: دکتر مهدی تاتاری

۲۴ مرداد ۱۴۰۴

۱ مقدمه

در این متن، به بررسی تکنیک‌هایی در مدل‌سازی معادله انتقال یک بعدی با شرایط اولیه و مرزی ناپیوسته با استفاده از شبکه‌های عصبی مطلع از فیزیک می‌پردازیم. به‌طور خاص، از ویژگی‌های فوریه تصادفی [۱]، آموزش نوبتی ویژگی‌های فوریه و شبکه، بهینه‌سازی با استفاده از الگوریتم LBFGS [۴] و وزن‌دهی تطبیقی بهره می‌بریم. این روش‌ها به‌طور مؤثری توانسته‌اند مدل‌ها را در برابر چالش‌های پیچیده مسائل دیفرانسیلی و بهینه‌سازی‌های مربوط به آن‌ها مقاوم کنند. شبکه‌های عصبی به‌عنوان یکی از قدرتمندترین ابزارهای یادگیری ماشینی [۲] شناخته می‌شوند و در سال‌های اخیر در مسائل متنوعی از بینایی ماشین تا پردازش زبان طبیعی کاربردهای موفقی نشان داده‌اند. در حوزهٔ مدل‌سازی سیستم‌های فیزیکی، ترکیب آموزه‌های فیزیک با ظرفیتِ تقریب‌پذیری شبکه‌های عصبی منجر به شاخه‌ای نوین به نام *شبکه‌های عصبی مطلع از فیزیک* [۱] شده است. هدف این رویکرد آن است که بدون اتکا صرف به داده‌های برچسب‌خورده، با وارد کردن معادلات بنیادی (معادلات با مشتقات جزئی، قوانین حفاظت، شرایط مرزی/ابتدایی) به فرایند یادگیری، تخمین‌های سازگار با فیزیک و در عین حال منعطف نسبت به پیچیدگی‌های مسئله به‌دست آورد.

۲ صورت کلی معادله‌ی انتقال

در این بخش، معادله انتقال یک‌بعدی به صورت کلی‌تر معرفی می‌شود که شامل ضریب انتقال متغیر $a(x,t)$ و ترم منبع $f(x,t)$ است:

$$\frac{\partial u}{\partial t} + a(x,t)\,\frac{\partial u}{\partial x} = f(x,t), \quad x \in [a,b], \quad t \in [0,T]$$

- $u(x,t)$: متغیر وابسته به مکان x و زمان t که نشان‌دهنده ویژگی‌ای است که در حال جابه‌جایی است، مانند غلظت آلودگی، دما یا سرعت جریان.
- $a(x,t)$: ضریب انتقال که متغیر است و می‌تواند بستگی به موقعیت مکان و زمان داشته باشد. این ضریب معمولاً نشان‌دهنده سرعت یا ویژگی حرکت ماده یا ویژگی دیگری در یک محیط است.
- $f(x,t)$: ترم منبع که نشان‌دهنده ورودی یا تولید به سیستم است، مانند منابع گرما، آلودگی یا مواد شیمیایی از منابع خارجی به سیستم.

شرایط اولیه و مرزی این معادله معمولاً به‌صورت زیر تعریف می‌شوند:

$$u(x,0) = u_0(x), \quad x \in [a,b],$$

[۱] Physics-Informed Neural Networks

$$u(x_0,t)=g(t) \quad \text{یا} \quad \frac{\partial u}{\partial x}(x_0,t)=h(t), \quad t\in[0,T].$$

در این‌جا $u_0(x)$ مقدار اولیه متغیر u را در زمان $t=0$ مشخص می‌کند و بسته به نوع مسأله فیزیکی، روی یکی از مرزهای مکانی $x=x_0$ با مقدار تابع u تعیین می‌شود (شرط دیریکله) یا مشتق فضایی آن (شرط نویمن).

۳ شبکه عصبی مطلع از فیزیک

شبکه‌های عصبی فیزیک آگاه [۵]، رویکردی نوین برای حل مسائل دیفرانسیل جزئی است. این شبکه‌ها با ترکیب یادگیری ماشینی و روابط فیزیکی ضمنی در مسئله، قادر هستند معادلات دیفرانسیل جزئی را به‌طور مؤثر حل کنند. در این روش، شبکه عصبی به‌عنوان تابع تقریب‌زننده $\hat{u}(x,t;\theta)$ عمل می‌کند که در آن θ پارامترهای شبکه عصبی هستند.

یک مسئله PDE به‌طور کلی به‌صورت زیر قابل تعریف است:

$$\mathcal{N}[u](\mathbf{x},t)=0, \quad \mathbf{x}\in\Omega,\ t\in[0,T],$$

$$u(\mathbf{x},t)=g(\mathbf{x},t) \quad \text{برای مرزهای مسئله.}$$

در این روابط:

- $\mathcal{N}$ عملگر دیفرانسیلی حاکم بر مسئله،
- u تابع مجهولی که هدف تقریب آن است،
- $\mathbf{x}\in\mathbb{R}^d$ متغیرهای فضایی،
- t متغیر زمانی،
- Ω ناحیه فضایی حل مسئله،
- g مقادیر مرزی یا اولیه مسئله،
- T زمان نهایی.

شبکه عصبی $\hat{u}(\mathbf{x},t;\theta)$ برای تقریب $u(\mathbf{x},t)$ استفاده می‌شود، به‌گونه‌ای که شرایط حاکم بر PDE و شرایط مرزی را برآورده سازد. این کار با تعریف یک تابع هزینه [۲] مناسب انجام می‌شود.

---

[۲] Loss-Function

تابع هزینه

تابع هزینه در شبکه‌های عصبی فیزیک آگاه شامل سه بخش اصلی است: ۱. خطای حاکم بر معادله دیفرانسیل: میزان انحراف تابع خروجی شبکه از معادلات دیفرانسیل جزئی. ۲. خطای شرایط اولیه: میزان انحراف تابع خروجی شبکه از شرایط اولیه. ۳. خطای شرایط مرزی: میزان انحراف تابع خروجی شبکه از شرایط مرزی.

تابع هزینه کلی به‌صورت زیر تعریف می‌شود:

$$\mathcal{L}(\theta) = \lambda_{\text{PDE}}\,\mathcal{L}_{\text{PDE}}(\theta) + \lambda_{\text{IC}}\,\mathcal{L}_{\text{IC}}(\theta) + \lambda_{\text{BC}}\,\mathcal{L}_{\text{BC}}(\theta),$$

که در آن:

- $\mathcal{L}_{\text{PDE}}(\theta)$: خطای ناشی از عدم تطابق مدل با معادله دیفرانسیل، به‌صورت:

$$\mathcal{L}_{\text{PDE}}(\theta) = \frac{1}{N_r}\sum_{i=1}^{N_r} \left|\mathcal{N}[\hat{u}](\mathbf{x}_i, t_i; \theta)\right|^2,$$

که در آن N_r تعداد نقاط حوزه و $\mathcal{N}[\hat{u}]$ خروجی معادله دیفرانسیل برای $\hat{u}$ است و اغلب توسط مشتق‌گیری خودکار معکوس محاسبه می‌شود. در معادله انتقال این خطا به شکل زیر نوشته می‌شود.

$$\mathcal{L}_{\text{PDE}}(\theta) = \frac{1}{N_r}\sum_{i=1}^{N_r} \left|\frac{\partial \hat{u}}{\partial x}(x_i, t_i, \theta) + a(x_i, t_i)\frac{\partial \hat{u}}{\partial t}(x_i, t_i, \theta) - f(x_i, t_i)\right|^2,$$

- $\mathcal{L}_{\text{IC}}(\theta)$: خطای ناشی از انحراف شرایط اولیه:

$$\mathcal{L}_{\text{IC}}(\theta) = \frac{1}{N_{\text{IC}}}\sum_{j=1}^{N_{\text{IC}}} \left|\hat{u}(\mathbf{x}_j, 0; \theta) - u_0(\mathbf{x}_j)\right|^2,$$

که در آن N_{IC} تعداد نقاط شرایط اولیه است و u_0 مقادیر مرزی اولیه می‌باشد.

- $\mathcal{L}_{\text{BC}}(\theta)$: خطای ناشی از انحراف شرایط مرزی:

$$\mathcal{L}_{\text{BC}}(\theta) = \frac{1}{N_{\text{BC}}}\sum_{k=1}^{N_{\text{BC}}} \left|\hat{u}(\mathbf{x}_k, t_k; \theta) - g(\mathbf{x}_k, t_k)\right|^2,$$

که در آن N_{BC} تعداد نقاط مرزی است و g مقادیر شرایط مرزی می‌باشد.

در این روش، شبکه عصبی سعی می‌کند توابعی را یاد بگیرد که هم معادلات دیفرانسیل و هم شرایط مرزی و اولیه داده شده را با کمترین خطا برآورده سازد. این امر با کمینه کردن تابع هزینه $\mathcal{L}(\theta)$ صورت می‌گیرد.

- بخش $\mathcal{L}_{\text{PDE}}$ تضمین می‌کند که خروجی شبکه تا حد امکان با معادله دیفرانسیل مطابقت داشته باشد. - بخش $\mathcal{L}_{\text{IC}}$ تضمین می‌کند که خروجی شبکه با شرایط اولیه مسئله سازگار باشد. - بخش $\mathcal{L}_{\text{BC}}$ تضمین می‌کند که خروجی شبکه با شرایط مرزی مسئله سازگار باشد.

مزایای شبکه‌های عصبی مطلع از فیزیک

- تقریب مؤثر: شبکه عصبی بدون نیاز به گسسته‌سازی پیچیده قادر به حل هاPDE در حوزه‌های چندبعدی است.
- انعطاف‌پذیری: می‌تواند برای انواع مختلفی از شرایط مرزی و معادلات غیرخطی استفاده شود.
- ترکیب داده و فیزیک: امکان استفاده از داده‌های تجربی و معادلات ریاضی به‌طور هم‌زمان برای حل مسائل.

۴ چند راهکار اولیه برای بهبود آموزش

۱.۴ ویژگی فوریه تصادفی

ویژگی فوریه تصادفی [۳] روشی مؤثر برای بهبود توانایی شبکه‌های عصبی در مدل‌سازی توابع با تغییرات سریع، ناپیوستگی‌های موضعی و مؤلفه‌های فرکانس بالا است. این تکنیک به منظور رفع سوگیری طیفی [۴] موجود در شبکه‌های چندلایه [۵] توسعه یافته است؛ سوگیری‌ای که موجب می‌شود مدل‌ها تمایل بیشتری به یادگیری مؤلفه‌های فرکانس پایین داشته باشند و بازتولید جزئیات پرنوسان را به تأخیر اندازند.

در این رویکرد، بردار ورودی $\mathbf{m} \in \mathbb{R}^d$ پیش از ورود به شبکه عصبی، تحت یک نگاشت غیرخطی قرار می‌گیرد که به صورت زیر تعریف می‌شود:

$$\gamma(\mathbf{m}) = \begin{bmatrix} \cos(\mathbf{Bm}) \\ \sin(\mathbf{Bm}) \end{bmatrix} \in \mathbb{R}^{2D}, \tag{۱.۴}$$

که در آن:

- D تعداد کلی جفت‌های سینوس_کسینوس نگاشت شده است.
- $\mathbf{B} \in \mathbb{R}^{D \times d}$ ماتریسی است که مقیاس و جهات فرکانسی را بر ورودی اعمال می‌کند. سطرهای $\mathbf{B}$ معمولاً به صورت تصادفی از یک توزیع نرمال $\mathcal{N}(0, \sigma^2)$ نمونه‌برداری می‌شوند. با افزایش مقدار σ شبکه توانایی بهتری در یادگیری اجزای پرنوسان دارد. هز چند که افزایش بیش‌از حد آن می‌تواند به ناپایداری و افزایش خطر بیش‌برازش منجر شود.

---

[۳] Random Fourier Feature

[۴] spectral bias

[۵] Multi Layer Perceptron

برای مثال در معادله انتقال تک بعدی داریم

$$\hat{u}(x,t;\theta) = N(\gamma(\mathbf{m});\theta), \quad \mathbf{m} = \begin{bmatrix} x \\ t \end{bmatrix}$$

تجربه نشان می‌دهد که استفاده از نگاشت فوریه منجر همگرایی سریع‌تر و پایدارتر بهینه‌سازهای نیوتونی و شبه نیوتونی مثل L-BFGS می‌شود. این امر به حدی مشهود است که گاهی از اوقات در مسائل غیردشوار، بهینه‌سازی مثل L-BFGS به تنهایی و با سرعت بالایی می‌تواند توابع بد رفتار را مدل کند. نگاشت حاصل، ورودی را به فضایی گسترش می‌دهد که در آن مؤلفه‌های فرکانس پایین و بالا به صورت خطی در دسترس شبکه قرار دارند. این ویژگی باعث می‌شود مدل بتواند بدون نیاز به افزایش عمق یا پیچیدگی معماری، ساختارهای محلی دقیق و ناپیوستگی‌های موجود در داده را به شکلی مؤثرتر مدل کند. نتیجه‌ی کار، بهبود قابل‌توجهی در کیفیت بازسازی، دقت پیش‌بینی و سرعت همگرایی حین آموزش شبکه است.

۲.۴ نمونه‌برداری تطبیقی

نمونه‌برداری تطبیقی یکی از روش‌های پیشرفته در بهینه‌سازی شبکه‌های عصبی مبتنی بر فیزیک است که با هدف بهبود دقت و کارایی مدل از انتخاب هوشمندانه نقاط نمونه‌برداری استفاده می‌کند. برخلاف نمونه‌برداری تصادفی که نقاط به صورت یکنواخت و بدون در نظر گرفتن ویژگی‌های مسئله انتخاب می‌شوند، نمونه‌برداری تطبیقی با استفاده از اطلاعات به‌دست‌آمده در طول فرآیند آموزش، نقاط جدید را به طور هدفمند انتخاب می‌کند. با تمرکز بر نقاطی که مدل در آن‌ها عملکرد ضعیفی دارد، دقت کلی مدل بهبود می‌یابد. انتخاب هوشمندانه نقاط باعث بهینه‌سازی استفاده از منابع محاسباتی می‌شود و مدل قادر است الگوهای پیچیده‌تر را بهتر یاد بگیرد و در نتیجه تعمیم‌پذیری بیشتری داشته باشد. این فرآیند به این صورت است که پس از هر دوره آموزشی، نقاطی که بیشترین باقیمانده را دارند شناسایی می‌شوند. نقاط جدید به گونه‌ای انتخاب می‌شوند که بیشتر در مناطق با خطای بالا قرار گیرند. مجموعه نقاط نمونه‌برداری با نقاط جدید به‌روز می‌شود و فرآیند آموزش ادامه می‌یابد.

۳.۴ وزن‌دهی تطبیقی در آموزش مدل‌های شبکه‌های عصبی

یکی از معضلات شبکه‌های عصبی فیزیک محور توازن در بخش‌های مختلف تابع هزینه است. خیلی از اوقات ممکن است در طر فرآیند آموزش یک بخش از تابع هزینه برخلاف بخش‌های دیگر با کاهش جزئی همراه باشد که منجر به عدم توانایی مدل در یادگیری جواب خواهد شد. در مقالات متعدد از روش‌هایی براساس استفاده از میزان خطا، نرم گرادیان و عناصر ماتریس NTK برای بخش‌های مختلف تابع هزینه ارائه شده است.

۱.۳.۴ وزن‌دهی تطبیقی بر اساس زیان هر بخش

$$\lambda_i = \frac{\frac{N \cdot f(\mathcal{L}_i)}{\sum_{j=1}^{N} f(\mathcal{L}_j)} + \beta}{1 + \beta}$$

که در آن:

- $\mathcal{L}_i$ زیان هر بخش است
- $f(\mathcal{L}_i)$ نتیجه اعمال تابع f بر روی ورودی $\mathcal{L}_i$ است.
- N تعداد ورودی‌هاست.
- $\sum_{j=1}^{N} f(\mathcal{L}_j)$ مجموع خروجی‌های اعمال‌شده توسط تابع f روی تمامی ورودی‌هاست.
- β مقدار بایاس ثابت است که برای جلوگیری از نوسانات و تقسیم‌بر_صفر به تابع اضافه می‌شود.

همچنین می‌توانیم برای نرم‌تر کردن وزن‌دهی و کاهش حساسیت وزن‌دهی نسبت به تابع f مقدار β را بعد از تعداد مشخصی ایپاک افزایش دهیم.

$$\beta_i = \beta_{i-1} + \varepsilon$$

این روش می‌تواند به‌سادگی با استفاده از یک تابع خطی یا درجه دوم برای f پیاده‌سازی شود تا بسته به نیاز، وزن‌دهی‌های متفاوتی ایجاد شود. در پیاده‌سازی انتخاب $f(x) = x$ و $f(x) = x^2$ پیشنهاد می‌شود. این تنظیمات باعث می‌شود که مجموع ضرایب تابع هزینه همواره برابر با تعداد بخش‌های تابع هزینه شود. به‌عبارت‌دیگر، خروجی وزن‌دهی تطبیقی به گونه‌ای تنظیم می‌شود که مقادیر به‌درستی توزیع شده و نوسانات ناشی از مقیاس‌های مختلف کاهش یابند.

۲.۳.۴ وزن‌دهی تطبیقی بر اساس گرادیان

یکی از روش‌های وزن‌دهی در شبکه‌های عصبی فیزیک محور، استفاده از **نرمال‌سازی گرادیان‌ها** برای تنظیم اهمیت هر بخش از تابع هزینه است. در این روش، گرادیان هر بخش از زیان نسبت به پارامترهای شبکه محاسبه می‌شود و وزن‌دهی بر اساس بزرگی گرادیان‌ها انجام می‌گیرد. ایده اصلی این است که بخش‌هایی از تابع هزینه که گرادیان بزرگ‌تری دارند، معمولاً سهم بیشتری در به‌روزرسانی پارامترها دارند و برعکس بخش‌هایی که گرادیان کوچک دارند، وزن کمتری به آن‌ها اختصاص می‌یابد.

وزن‌دهی تطبیقی بر اساس گرادیان معمولاً به شکل زیر تعریف می‌شود:

$$\lambda_i = \frac{\frac{1}{\|\nabla_\theta \mathcal{L}_i\| + \varepsilon}}{\sum_{j=1}^{N} \frac{1}{\|\nabla_\theta \mathcal{L}_j\| + \varepsilon}} \cdot N$$

که در آن:

- $\mathcal{L}_i$ زیان مربوط به بخش i است.
- $\nabla_\theta \mathcal{L}_i$ گرادیان آن زیان نسبت به پارامترهای شبکه است.
- ε مقدار بسیار کوچک برای جلوگیری از تقسیم بر صفر است.
- N تعداد بخش‌های تابع هزینه است تا مجموع وزن‌ها نرمال و قابل مقایسه باشد.

۳.۳.۴ وزن‌دهی تطبیقی بر اساس ماتریس NTK

روش دیگر وزن‌دهی تطبیقی در شبکه‌های عصبی فیزیک محور، استفاده از **ماتریس NTK** [۶] است. NTK به عنوان یک تقریب خطی از تغییرات خروجی شبکه نسبت به پارامترها در نزدیکی نقطه فعلی بهینه‌سازی عمل می‌کند و ساختار حساسیت شبکه را نسبت به هر بخش تابع هزینه نشان می‌دهد. به بیان دقیق‌تر، اگر θ بردار پارامترهای شبکه باشد و $U(\mathbf{x};\theta)$ خروجی اسکالر شبکه در ورودی $\mathbf{x}$، هر درایه Θ_{lk} از ماتریس Θ برای یک مجموعه ورودی $\{\mathbf{x}_k\}_{k=1}^{M}$ به صورت زیر تعریف می‌شود

$$\Theta_{lk} = \langle \nabla_\theta U(\mathbf{x}_l;\theta), \nabla_\theta U(\mathbf{x}_k;\theta) \rangle$$

که نشان‌دهنده میزان تأثیر تغییرات پارامترها بر خروجی‌های مربوط به نقاط $\mathbf{x}_k$ و $\mathbf{x}_l$ است.

وزن‌دهی هر بخش می‌تواند بر اساس معیارهایی مانند نُرم فرُبنیوس $\|\Theta_i\|_F$ یا $\text{trace}(\Theta_i)$ انجام شود. هر چند استفاده از $\|\Theta_i\|_F$ دقت بیشتری دارد، به دلیل هزینه بالای محاسبه نُرم فرُبنیوس، معمولاً از $\text{trace}(\Theta_i)$ استفاده می‌شود. اگر Θ_i ماتریس NTK مربوط به بخش‌های مختلف تابع زیان در نقاط مربوط به آن باشد، با نرمال کردن، وزن بخش i به صورت زیر محاسبه می‌شود

$$\lambda_i = \frac{\text{trace}(\Theta_i)}{\sum_{j=1}^{N} \text{trace}(\Theta_j)} N$$

---

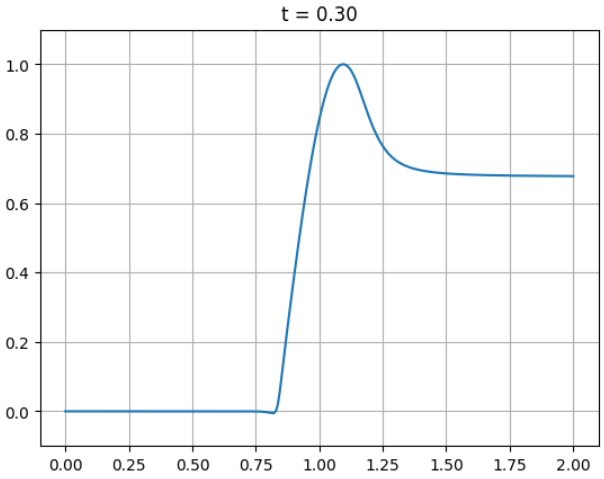
[۶] Neural Tangent Kernel

۵ شروع با یک مسئله ساده

در این بخش با یک مسئله معادله انتقال ساده شروع می‌کنیم و خواننده را با چالش‌های پیش رو آشنا خواهیم کرد. برای شروع به حل معادله‌ی

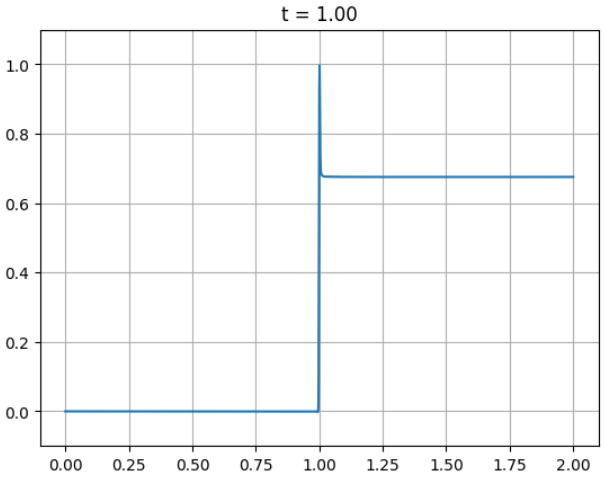
$$\frac{\partial u}{\partial t} - 6(x-1)\frac{\partial u}{\partial x} = 0, \quad u(x,0) = \sin x, \quad u(0,t) = 0, \quad x \in [0,\pi], \quad t \in [0,1]$$

می‌پردازیم.



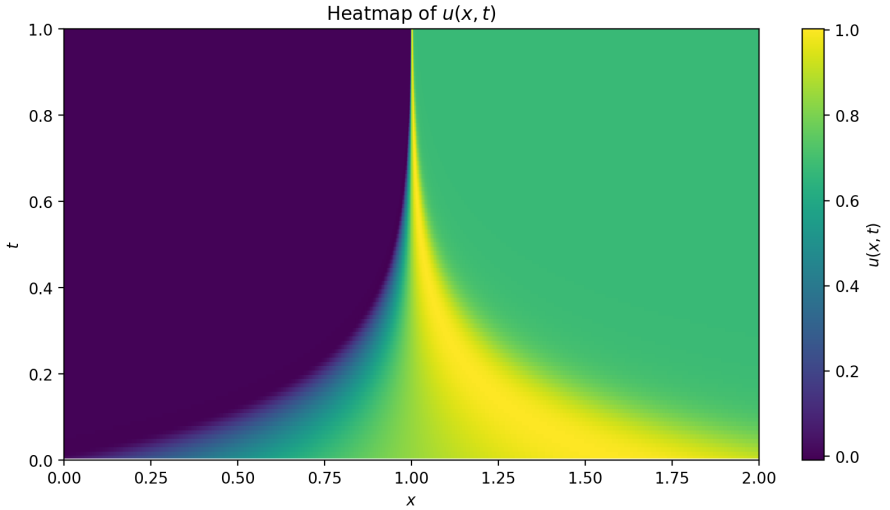
شکل ۱.۵: تصویر اول

شکل ۲.۵: تصویر دوم



شکل ۳.۵: تصویر سوم

شکل ۴.۵: تصویر چهارم



شکل ۵.۵: تصویر پنجم

۶ مشتق‌گیری عددی

هر چند مشتق‌گیری خودکار برتری خاصی نسبت به روش‌های عددی در مشتق‌گیری دارد، برخی اوقات استفاده از روش‌های عددی مشتق‌گیری می‌تواند کمک کننده باشد. در استفاده از رابطه مرکزی مرتبه دوم برای تقریب مشتق داریم

$$\frac{\partial \hat{u}}{\partial x}(x_i,t_i;\theta)=\frac{\hat{u}(x_i+h,t_i;\theta)-\hat{u}(x_i-h,t_i;\theta)}{2h}-\frac{h^2}{6}\frac{\partial^3 \hat{u}}{\partial x^3}(\xi(x_i),t_i;\theta), \tag{۱.۶}$$

$$\frac{\partial \hat{u}}{\partial t}(x_i,t_i;\theta)=\frac{\hat{u}(x_i,t_i+h;\theta)-\hat{u}(x_i,t_i-h;\theta)}{2h}-\frac{h^2}{6}\frac{\partial^3 \hat{u}}{\partial t^3}(x_i,\xi(t_i);\theta) \tag{۲.۶}$$

که در آن $\xi(x_i)\in[x_i-h,x_i+h]$ و $\xi(t_i)\in[t_i-h,t_i+h]$. بنابراین $\tilde{\mathcal{L}}_{\text{PDE}}(\theta)$ که خطای جدید مربوط به مانده‌ی معادله است برابر است با

$$\frac{1}{N_r}\sum_{i=1}^{N_r}\left|\frac{\hat{u}(x_i,t_i+h;\theta)-\hat{u}(x_i,t_i-h;\theta)}{2h}+a(x_i,t_i)\frac{\hat{u}(x_i+h,t_i;\theta)-\hat{u}(x_i-h,t_i;\theta)}{2h}-f(x,t)\right|^2$$

فرض کنید $\hat{\Omega}$ مجموعه‌ای دلخواه از نقاط در Ω باشد. تعریف می‌کنیم

$$\ell_{\text{PDE}}(\theta)=\max_{x\in\hat{\Omega}}\left|\frac{\partial \hat{u}}{\partial t}(x_i,t_i;\theta)+a(x_i,t_i)\frac{\partial \hat{u}}{\partial x}(x_i,t_i;\theta)-f(x,t)\right|$$

$$\tilde{\ell}_{\text{PDE}}(\theta)=\max_{x\in\hat{\Omega}}\left|\frac{\hat{u}(x_i,t_i+h;\theta)-\hat{u}(x_i,t_i-h;\theta)}{2h}+a(x_i,t_i)\frac{\hat{u}(x_i+h,t_i;\theta)-\hat{u}(x_i-h,t_i;\theta)}{2h}-f(x,t)\right|$$

با استفاده از روابط (۱.۶، ۲.۶) به سادگی می‌توان دید

$$\tilde{\ell}_{\text{PDE}}(\theta)\le\ell_{\text{PDE}}(\theta)+\frac{h^2}{6}\max_{x\in\hat{\Omega}}\left|\frac{\partial^3 \hat{u}}{\partial t^3}(x_i,\xi(t_i);\theta)+a(x_i,t_i)\frac{\partial^3 \hat{u}}{\partial x^3}(\xi(x_i),t_i;\theta)\right|$$

بنابراین به طور واضح

$$\left|\sqrt{\tilde{\mathcal{L}}_{\text{PDE}}(\theta)}-\sqrt{\mathcal{L}_{\text{PDE}}(\theta)}\right|\le\left|\tilde{\ell}_{\text{PDE}}(\theta)-\ell_{\text{PDE}}(\theta)\right|\le\frac{h^2}{6}\left(\max_{x\in\Omega}\left|\frac{\partial^3 \hat{u}}{\partial t^3}(x,t;\theta)\right|+M\max_{x\in\Omega}\left|\frac{\partial^3 \hat{u}}{\partial x^3}(x,t;\theta)\right|\right)$$

که $M=\max_{x\in\Omega}|a(x,t)|$.

علاوه بر طول گام h، کوچک بودن مقدار مشتق سوم $\hat{u}(x,t;\theta)$ نسبت به x و t در طول فرآیند آموزش باعث کاهش مقدار

$\left|\tilde{\ell}_{\text{PDE}}(\theta) - \ell_{\text{PDE}}(\theta)\right|$ می‌شود. در استفاده از رابطه مرکزی مرتبه 4 برای تقریب مشتق می‌توان نشان داد

$$\left|\tilde{\ell}_{\text{PDE}}(\theta) - \ell_{\text{PDE}}(\theta)\right| \leq \frac{h^4}{30}\left(\max_{x\in\Omega}\left|\frac{\partial^5 \hat{u}}{\partial t^5}(x,t;\theta)\right| + M\max_{x\in\Omega}\left|\frac{\partial^5 \hat{u}}{\partial x^5}(x,t;\theta)\right|\right)$$

بنابراین در برخورد با توابع نرم و خوش رفتار مشتق‌گیری عددی راهکاری مناسب و کم هزینه‌تر نسبت به مشتق‌گیری خودکار است. هر چند اغلب محاسبات در سیستم ممیزشناور 32 بیتی صورت می‌گیرد و در کوچک‌کردن مقدار h محدودیت وجود دارد، بزرگ بودن h می‌تواند در جلوگیری از بیش‌برازش کمک‌کننده باشد. برای مثال در استفاده از رابطه مرکزی مرتبه 2 و مرتبه 4 به ازای هر نقطه کالوکیشن در محاسبه مشتق $\hat{u}(x_i,t_i;\theta)$، به ترتیب 2 و 4 نقطه در فرآیند آموزش درگیر می‌شوند. لازم به ذکر است در برخی موارد استفاده از مشتق‌گیری عددی می‌تواند در سرعت همگرایی بهینه‌ساز کمک کننده باشد. این روش به‌خصوص در معادلاتی که دارای مشتقات مرتبه دوم و بالاتر است کاربرد بیشتری دارد؛ چرا که گراف محاسباتی حاصل از مشتقات مرتبه بالا اغلب پیچیده و دارای حجم بالایی می‌باشد. می‌توان در مراحل ابتدایی آموزش از مشتق‌گیری عددی و مراحل انتهایی از مشتق‌گیری خودکار استفاده کرد.

۷ تحمیل شرایط اولیه و مرزی به عنوان یک قید سخت

گاهی اوقات به جای کار با یک شبکه عصبی که شرایط اولیه، مرزی و معادله حاکم بر مسئله را یاد بگیرد، از یک جواب اولیه استفاده می‌کنیم که شرایط اولیه یا مرزی یا هر دوی آن‌ها را برآورده کند. به عبارتی

$$\hat{u}(x,t;\theta) = A(x,t) + B(x,t)N(x,t;\theta)$$

که در آن $N(x,t;\theta)$ شبکه عصبی با پارامترهای آموزشی θ است. $A(x,t)$ شرایط اولیه و شرایط مرزی را تأمین می‌کند و $B(x,t)$ در ناحیه مرزی و اولیه مقدار صفر دارد. برای مثال یکی از جواب‌های اولیه که شرایط اولیه و مرزی را تأمین می‌کند به شکل زیر است

$$\hat{u}(x,t;\theta) = t(x-x_0)N(x,t;\theta) + u(x_0,t) + u(x,0) - u(x_0,0)$$

و بنابراین

$$\frac{\partial \hat{u}}{\partial x}(x,t;\theta) = t\left[(x-x_0)\frac{\partial N}{\partial x}(x,t;\theta) + N(x,t;\theta)\right] + \frac{\partial u}{\partial x}(x,0) \qquad (۱.۷)$$

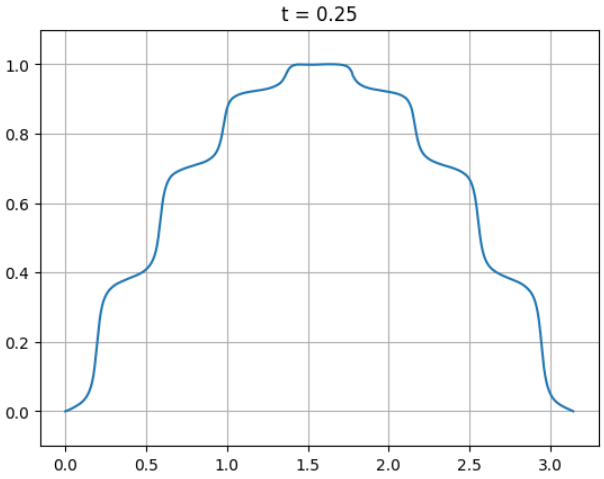
$$\frac{\partial \hat{u}}{\partial t}(x,t;\theta) = (x-x_0)\left[t\frac{\partial N}{\partial t}(x,t;\theta) + N(x,t;\theta)\right] + \frac{\partial u}{\partial t}(x_0,t) \qquad (۲.۷)$$

یکی از برتری‌های این روش این است که تابع هزینه تنها دارای یک بخش مربوط به مانده معادله را دارد و نیازی به آموزش شرایط اولیه و مرزی نیست. این در حالی است که بدرفتاری توابع $\frac{\partial u}{\partial x}(x,0)$ و $\frac{\partial u}{\partial t}(x_0,t)$ در رابطه‌های (۱.۷ و ۲.۷) برخی اوقات باعث افزایش بیش از حد تابع هزینه می‌شود و فرآیند آموزش را دچار اختلال می‌کنند. همچنین در این حالت مدل در شبیه‌سازی رفتارهای تند و نوسانی در نزدیکی $t = 0$ و $x = x_0$ دچار محدودیت است. بنابراین در برخود با مسائلی که رفتار مشتق شرایط اولیه و مرزی کنترل‌شده نباشد، این روش پیشنهاد نمی‌شود. واضح است که استفاده از این روش در مسائلی با شرایط اولیه یا مرزی ناپیوسته، بلااستفاده است.

در ادامه به حل مسئله

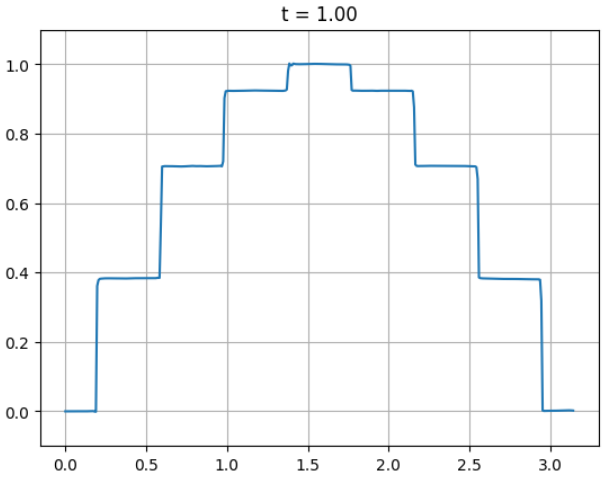
$$u_t + 0.5\sin(16x)u_x = 0, \quad u(x,0) = \sin(x), \quad u(0,t) = 0, \quad x \in [0,\pi], \quad t \in [0,1]$$

پنج شکل اول بیانگر جواب $\hat{u}(x,t;\theta^*)$ و پنج شکل دوم نشان‌دهنده‌ی $\hat{N}(x,t;\theta^*)$ است که θ^* پارامترهای آموزش دیده شبکه عصبی است.



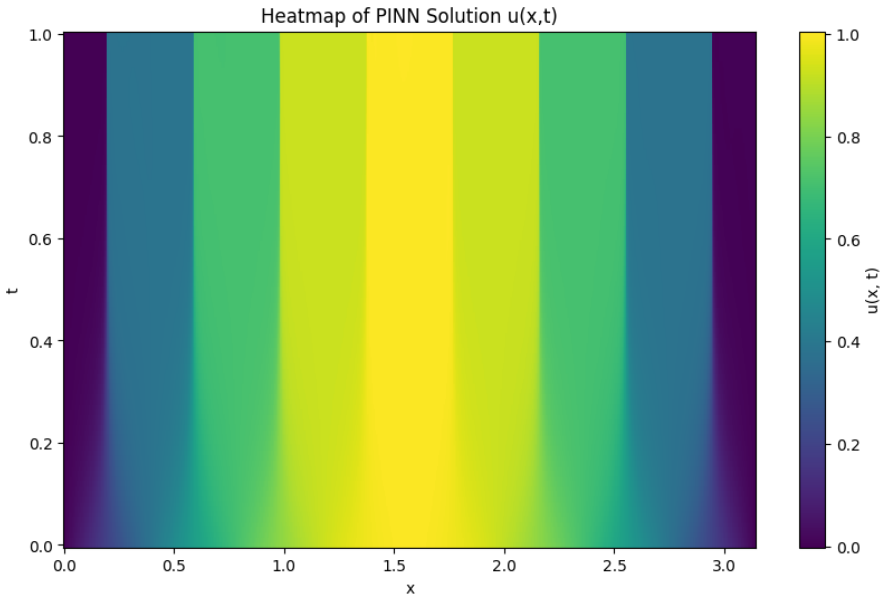
شکل ۱.۷: تصویر اول

شکل ۲.۷: تصویر دوم

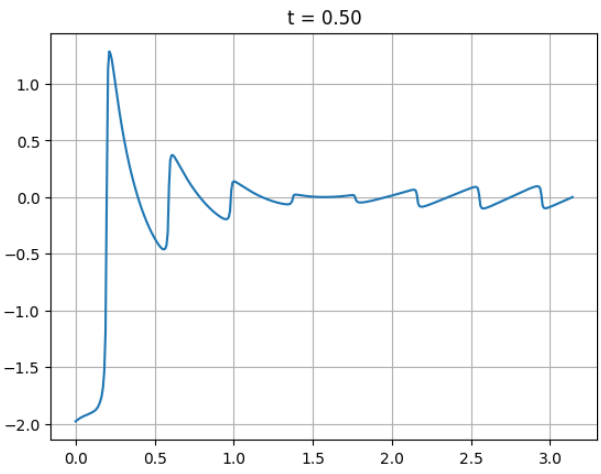


شکل ۳.۷: تصویر سوم

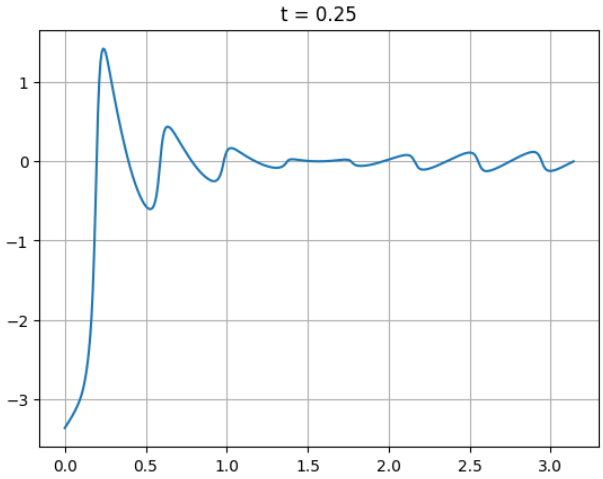
شکل ۴.۷: تصویر چهارم



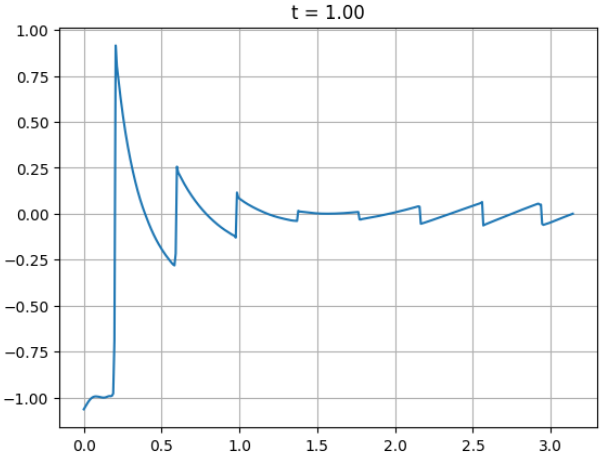
شکل ۵.۷: تصویر پنجم



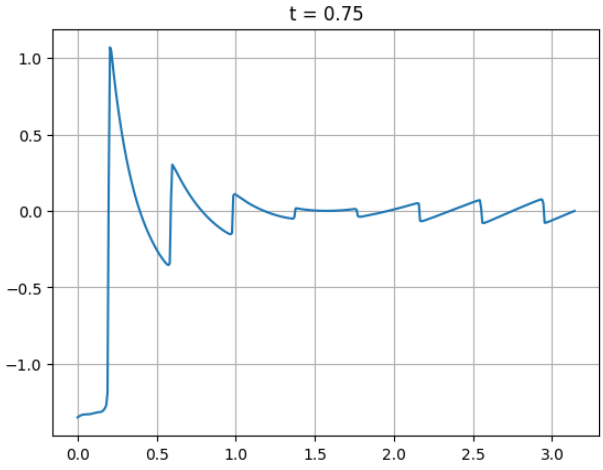
شکل ۷.۷: تصویر دوم



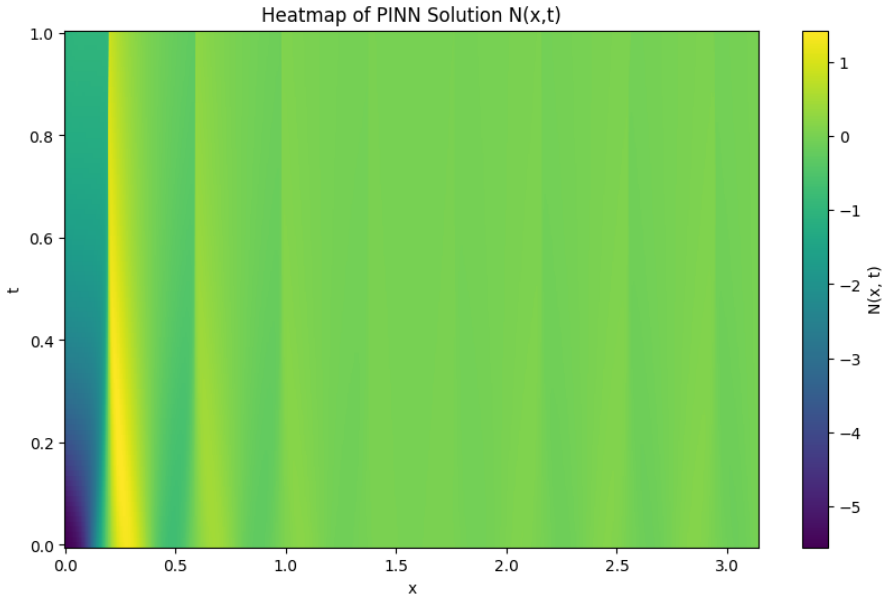
شکل ۶.۷: تصویر اول



شکل ۹.۷: تصویر چهارم



شکل ۸.۷: تصویر سوم



شکل ۱۰.۷: تصویر پنجم

۸ مرحله‌بندی آموزش

در برخی مسئله‌ها یادگیری شرایط اولیه و مرزی حاکم بر مسئله دشوار است و این موضوع باعث سخت‌تر شدن فرآیند آموزش، احتمال گیرافتادن در بهینه محلی و توجه کمتر بهینه‌ساز به بخش مربوط به باقی‌مانده معادله شود. در این حالت پیشنهاد می‌شود در طول آموزش مسئله با شرایط اولیه و مرزی ساده‌تر حل شود و به مرور زمان شرایط اولیه و مرزی به شرایط مرزی و اولیه واقعی نزدیک شود. برای مثال می‌توانیم شرایط اولیه و مرزی را توابع قطعه‌ای خطی با نقاط با فاصله مساوی تقریب بزنیم؛ به طوری که تقریب را با تعداد نقاط اندک شروع کنیم و به مدل اجازه دهیم تا جواب مسئله را با شرایط اولیه و مرزی ساده‌تر تا حد خوبی یاد بگیرد؛ پس از آن با افزایش تعداد نقاط شرایط اولیه و مرزی به تقریب بهتری از شرایط اولیه و مرزی می‌رسیم و تابع هزینه را به روز می‌کنیم. در نهایت می‌توان در تکرارهای آخر شرایط اولیه و مرزی اصلی را در تابع هدف قرار داد. این روش باعث کاهش احتمال گیر افتادن در بهینه محلی و توازن نسبی بخش‌های مختلف تابع هزینه در طول آموزش می‌شود. پنج شکل اول بیانگر جواب معادله

$$u_t + (1+x)u_x = 0, \quad t \in [0,1], \quad x \in [0,2]$$

با شرایط اولیه و مرزی

$$u(x,0) = \sin(x), \quad u(0,t) = t\sin^2(40t^2)$$

و پنج شکل دوم بیانگر جواب معادله

$$u_t - 6(x-0.5)(x-1)(x-1.5)u_x = 0, \quad t \in [0,1], \quad x \in [0,2]$$

با شرایط اولیه و مرزی

$$u(x,0) = \sin^2(4x), \quad u(0,t) = 0$$

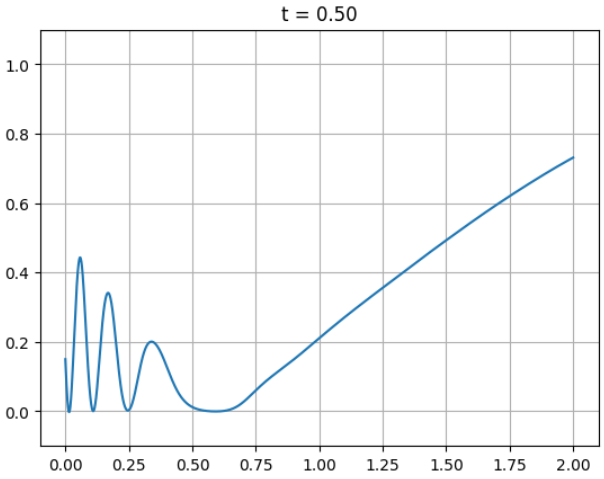
و پنج شکل سوم نشان‌دهنده‌ی جواب معادله‌ی

$$u_t - 5(x-0.5)(x-1)(x-1.5)u_x = 0, \quad t \in [0,1], \quad x \in [0,2]$$

با شرایط اولیه و مرزی

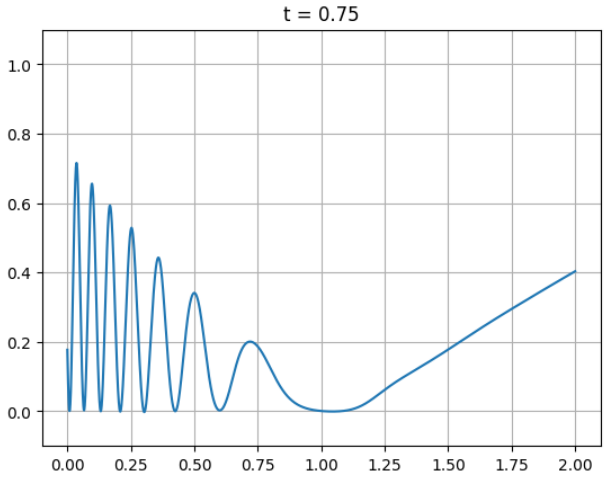
$$u(x,0) = \sin^2(8x), \quad u(0,t) = 0$$

است.

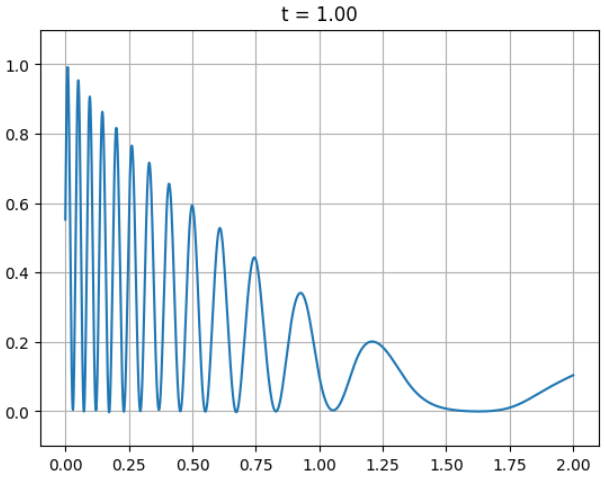


شکل ۱.۸: تصویر اول

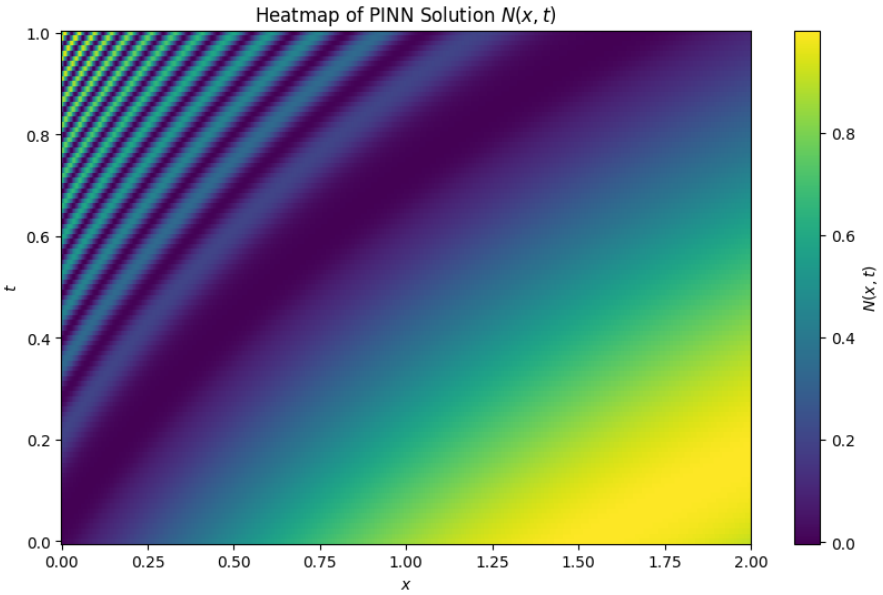
شکل ۲.۸: تصویر دوم



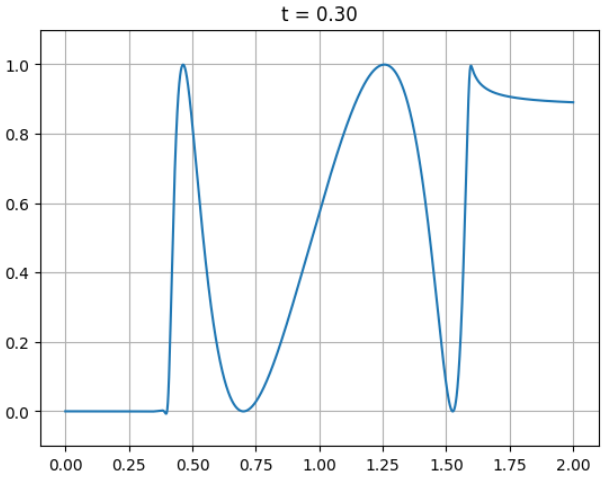
شکل ۳.۸: تصویر سوم



شکل ۴.۸: تصویر چهارم

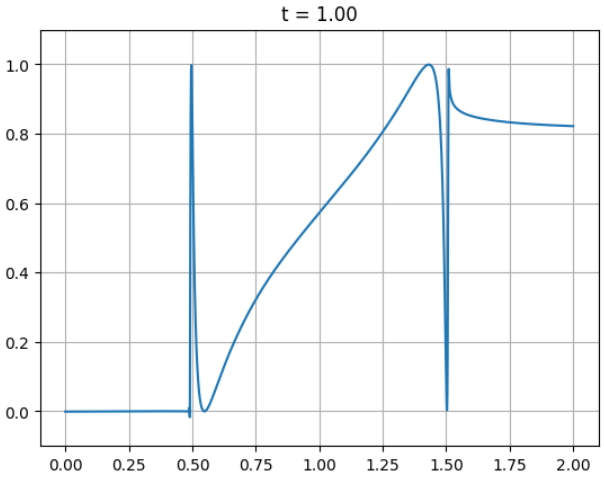


شکل ۵.۸: تصویر پنجم



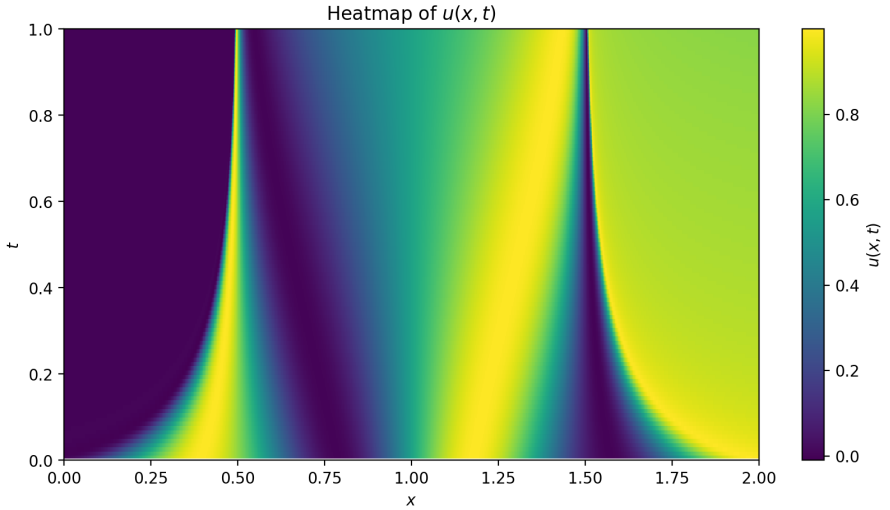
شکل ۶.۸: تصویر اول

شکل ۷.۸: تصویر دوم

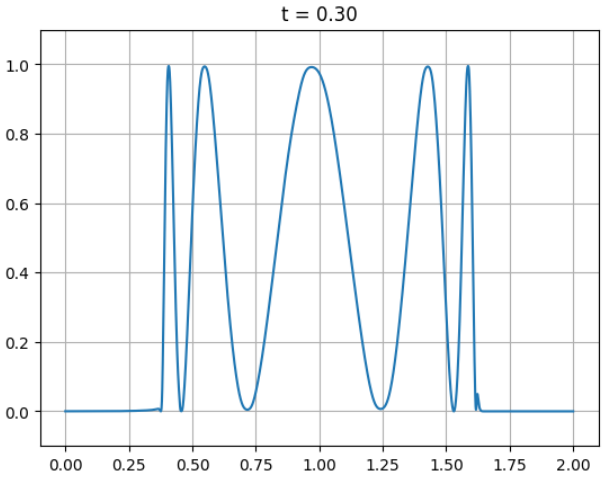


شکل ۸.۸: تصویر سوم

شکل ۹.۸: تصویر چهارم

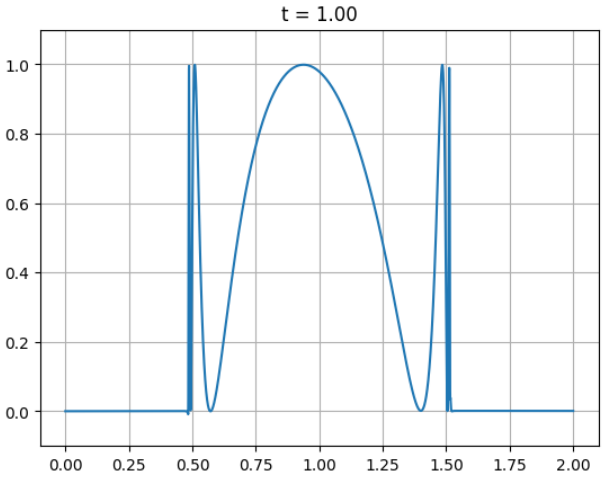


شکل ۱۰.۸: تصویر پنجم



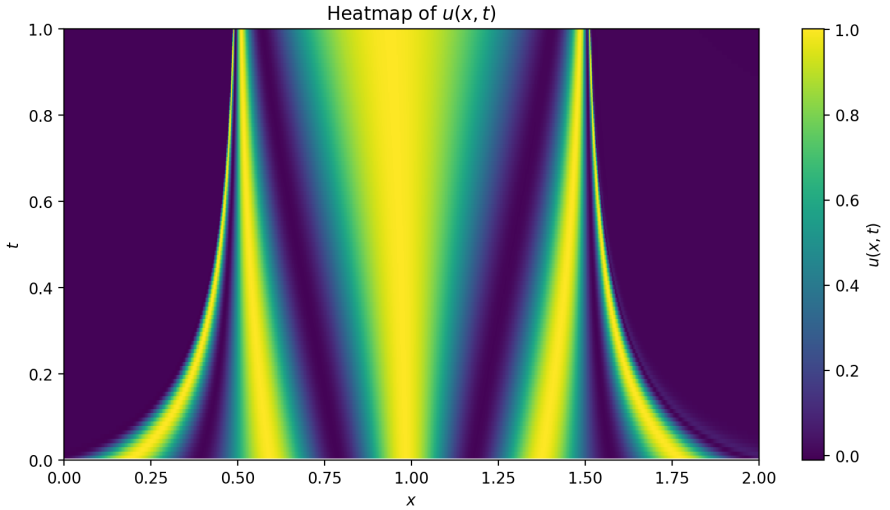
شکل ۱۱.۸ : تصویر اول

شکل ۱۲.۸ : تصویر دوم



شکل ۱۳.۸ : تصویر سوم

شکل ۱۴.۸ : تصویر چهارم



شکل ۱۵.۸ : تصویر پنجم

۹ نحوه برخورد با شرایط اولیه و مرزی ناپیوسته

هنگامی که شرایط اولیه یا مرزی ناپیوستگی دارند یکی از معضلات بزرگ کاهش زیان مربوط به این بخش‌ها و توازن بخش‌های مختلف تابع زیان است. بنابراین نیاز به رویکردی که داریم که این مشکل را برطرف کند.

۱.۹ بهینه سازی

در این بخش به بررسی تکنیک‌های بهینه‌سازی مورد استفاده در مدل‌سازی ناپیوستگی‌ها می‌پردازیم. بهینه‌سازی نقش حیاتی در یافتن پارامترهای بهینه شبکه عصبی و تضمین عملکرد مدل در حفظ ناپیوستگی‌ها دارد.

هرچند استفاده از نگاشت ویژگی فوریه منجر به بهبود سرعت بهینه‌سازی و نتایج دقیق‌تر خروجی می‌شود، نتایج تجربی نشان می‌دهد که آموزش نوبتی به ترتیب پارامترهای نگاشت فوریه و شبکه‌عصبی منجر به مدل‌سازی سریعتر ناپیوستگی‌ها می‌شود. بنابراین پیشنهاد می‌شود در ابتدای فرآیند یادگیری پارامترهای شبکه‌ی عصبی قفل شوند و با بهینه‌ساز L-BFGS پارامترهای نگاشت فوریه آموزش ببینند. پس از آن می‌توان پارامترهای نگاشت فوریه را قفل کرد و پارامترهای شبکه‌ی عصبی را آموزش داد. آموزش پارامترهای شبکه اغلب به این شکل انجام می‌شود که مراحل ابتدایی آموزش را با استفاده از یک بهینه‌ساز پایدار مبتنی بر گرادیان مثل Adam، آموزش را انجام می‌دهیم و مراحل انتهایی آموزش را با استفاده از بهینه‌ساز L-BFGS انجام می‌دهیم. بنابراین فرآیند آموزش به دو بخش تبدیل می‌شود.

۱. (مرحله اول آموزش پارامترهای فوریه) در این مرحله، عناصر $\mathbf{B}$ در رابطه (۱.۴) که نگاشت فوریه را تعریف می‌کنند، به‌طور جداگانه آموزش داده می‌شوند. هدف این است که این ویژگی‌ها بتوانند نمایندگی مناسبی از داده‌های ورودی فراهم کنند تا شبکه عصبی بتواند به بهترین نحو آنها را پردازش کند. لازم به ذکر است که افزایش بیش از حد عناصر ماتریس $\mathbf{B}$ می‌تواند به منجر به بیش‌برازش و همچنین ناپایداری شود.

۲. (مرحله دوم آموزش شبکه عصبی) پس از تثبیت نگاشت فوریه، پارامترهای شبکه عصبی θ به‌طور جداگانه آموزش داده می‌شوند. در این مرحله، نگاشت فوریه به‌دست‌آمده به عنوان ورودی به شبکه عصبی وارد شده و شبکه سعی می‌کند با استفاده از آنها خروجی‌های دقیق‌تری را تولید کند.

به عبارتی اگر

$$\hat{u}(x,t;\theta) = N(\gamma(x,t,\tilde{\theta});\theta)$$

که θ پارامترهای شبکه‌عصبی و $\tilde{\theta}$ پارامترهای آموزش‌پذیر فوریه است، ابتدا $\tilde{\theta}$ و پس از آن θ وارد فرآیند بهینه‌سازی می‌شود. این رویکرد باعث می‌شود که شبکه عصبی بتواند به‌طور مؤثری از نگاشت فوریه استفاده کند و عملکرد کلی مدل بهبود یابد. همچنین در مرحله آموزش ویژگی فوریه، وزن‌دهی تطبیقی نقش مهمی ایفا می‌کند. با تغییر بهینه فضای ویژگی در آموزش اولیه، وزن‌دهی تطبیقی کمک می‌کند تا ویژگی‌های فوریه به‌طور موثری برای بهینه‌سازی‌های بعدی تغییر کنند. این تغییرات باعث بهبود فضای ویژگی شده و سرعت همگرایی مدل را در مراحل بعدی آموزش افزایش می‌دهد.

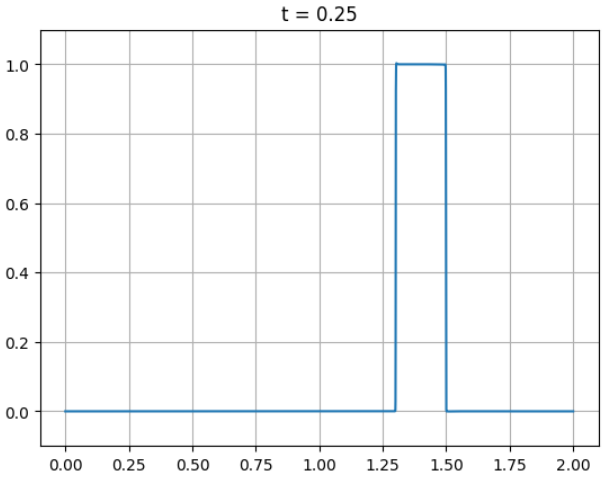
به‌طور خاص، آموزش اولیه ویژگی‌های فوریه با وزن‌دهی تطبیقی می‌تواند باعث سازگاری بیشتر فضای ویژگی با نیازهای خاص مدل شود. این تنظیمات همچنین احتمال گیرکردن مدل در بهینه‌های محلی را کاهش‌دهند و فرایند آموزش را به سمت راه‌حل‌های بهینه‌تری سوق می‌دهد. پنج شکل اول مربوط به معادله‌ی $u_t + 2u_x = 0$ با شرایط اولیه و مرزی

$$u(0,t) = 0, \quad u(x,0) = \begin{cases} 1 & |x - 0.9| \leq 0.1 \\ 0, & \text{در غیر این صورت} \end{cases}, \quad x \in [0,2], \quad t \in [0,1]$$

و پنج شکل دوم مربوط به معادله‌ی $u_t + 2u_x = 0$ با شرایط اولیه و مرزی

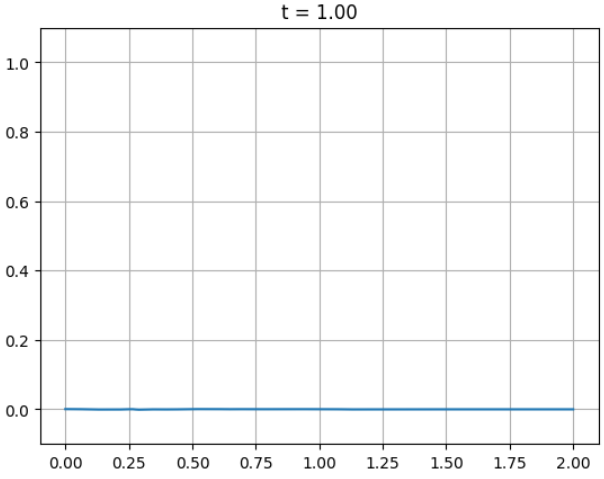
$$u(0,t) = \begin{cases} 0.5 & t \leq 0.5 \\ 0, & \text{در غیر این صورت} \end{cases}, \quad u(x,0) = \begin{cases} 1 & |x - 0.9| \leq 0.1 \\ 0, & \text{در غیر این صورت} \end{cases}, \quad x \in [0,2], \quad t \in [0,1]$$

است. واضح است که در این حالات همگرایی در نرم دو رخ می‌دهد و نه در نرم بی‌نهایت.



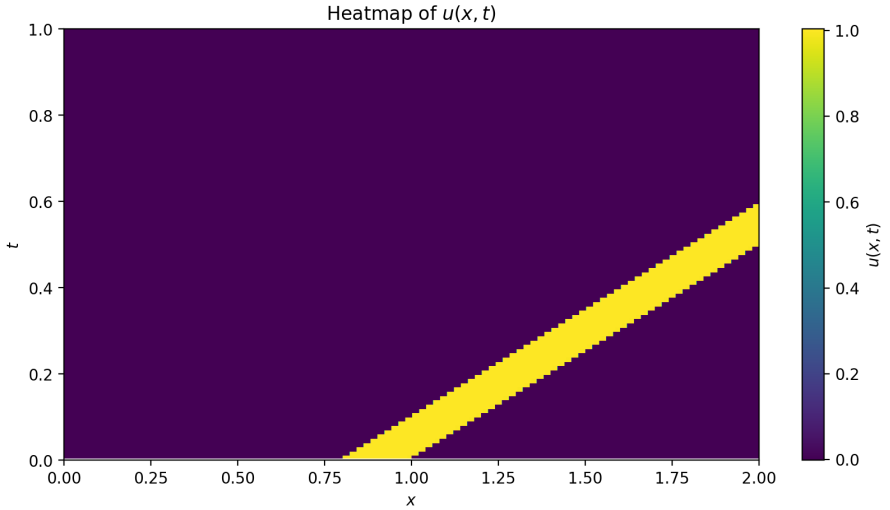
شکل ۱.۹: تصویر اول

شکل ۲.۹: تصویر دوم

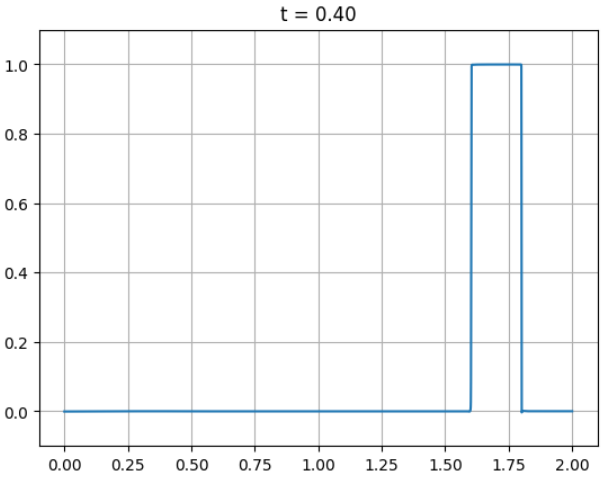


شکل ۳.۹: تصویر سوم

شکل ۴.۹: تصویر چهارم

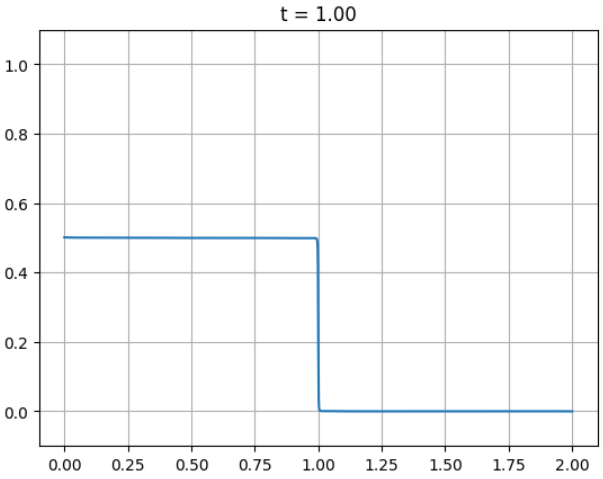


شکل ۵.۹: تصویر پنجم



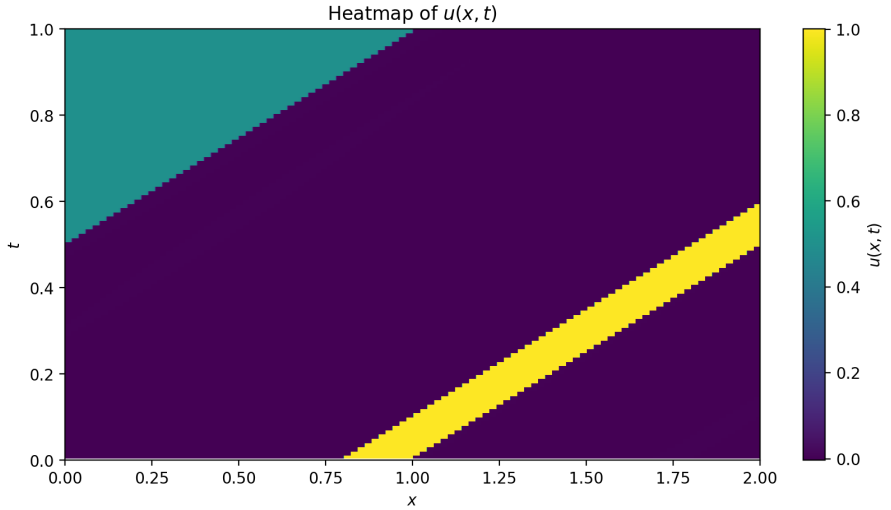
شکل ۶.۹: تصویر اول

شکل ۷.۹: تصویر دوم



شکل ۸.۹: تصویر سوم

شکل ۹.۹: تصویر چهارم



شکل ۱۰.۹: تصویر پنجم

مراجع

[1] A. Rahimi and B. Recht, “Random Features for Large-Scale Kernel Machines,” in *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, pp. ,1184–1177 2007.

[2] Y. Bengio, “Learning Deep Architectures for AI,” *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. ,127–1 2009. DOI: .1561/2200000006.10

[3] J. W. Tukey, *Exploratory Data Analysis*, Addison-Wesley, Reading, MA, 1977.

[4] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer, 2006. ISBN: -387-0-978 .1-30303

[5] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations,” *Journal of Computational Physics*, vol. 378, pp. ,707–686 2019. DOI: .045.10.2018.j.jcp/1016.10