Równania różniczkowe - spectral bias

Zadanie 1. Dane jest równanie różniczkowe zwyczajne

$$\frac{\mathrm{d}u(x)}{\mathrm{d}x} = \cos(\omega x) \text{ dla } x \in \Omega\,, \tag{1}$$

gdzie:
$x, \omega, u \in \mathbb{R}$,
x to położenie,
Ω to dziedzina, na której rozwiązujemy równanie, $\Omega = \{\, x \mid -2\pi \le x \le 2\pi \,\}$.
$u(\cdot)$ to funkcja, której postaci szukamy.

Warunek początkowy zdefiniowany jest następująco:

$$u(0) = 0\,. \tag{2}$$

Analityczna postać rozwiązania równania (1) z warunkiem początkowym (2) jest następująca:

$$u(x) = \frac{1}{\omega}\sin(\omega x)\,. \tag{3}$$

Rozwiąż powyższe zagadnienie początkowe (1,2). Do rozwiązania użyj sieci neuronowych PINN (ang. *Physics-informed Neural Network*) [1]. Można wykorzystać szablon w `pytorch`-u lub bibliotekę `DeepXDE` [2].
Koszt rezydualny zdefiniowany jest następująco:

$$\mathcal{L}_r(\theta) = \frac{1}{N}\sum_{i=1}^{N} || \frac{\mathrm{d}\hat{u}(x)}{\mathrm{d}x} - \cos(\omega x_i)||^2\,, \tag{4}$$

gdzie N jest liczbą punktów kolokacyjnych.
Koszt związany z warunkiem początkowym przyjmuje postać:

$$\mathcal{L}_{IC}(\theta) = ||\hat{u}(0) - 0||^2\,. \tag{5}$$

Funkcja kosztu zdefiniowana jest następująco:

$$\mathcal{L}(\theta) = \mathcal{L}_r(\theta) + \mathcal{L}_{IC}(\theta)\,. \tag{6}$$

Warstwa wejściowa sieci posiada 1 neuron, reprezentujący zmienną x, Warstwa wyjściowa także posiada 1 neuron, reprezentujący zmienną $\hat{u}(x)$. Uczenie trwa przez 50 000 kroków algorytmem Adam ze stałą uczenia równą 0.001. Jako funkcję aktywacji przyjmij tangens hiperboliczny, tanh.

(a) Przypadek $\omega = 1$.

Ustal następujące wartości:

- 2 warstwy ukryte, 16 neuronów w każdej warstwie
- liczba punktów treningowych: 200
- liczba punktów testowych: 1000

(b) Przypadek $\omega = 15$.

Ustal następujące wartości:

- liczba punktów treningowych: $200 \cdot 15 = 3000$
- liczba punktów testowych: 5000

Eksperymenty przeprowadź z trzema architekturami sieci:

- 2 warstwy ukryte, 16 neuronów w każdej warstwie
- 4 warstwy ukryte, 64 neurony w każdej warstwie
- 5 warstw ukrytych, 128 neuronów w każdej warstwie

(c) Dla wybranej przez siebie sieci porównaj wynik z rozwiązaniem, w którym przyjęto, że szukane rozwiązanie (*ansatz*) ma postać:

$$\hat{u}(x;\theta) = \tanh(\omega x) \cdot NN(x;\theta)\,. \tag{7}$$

Taka postać rozwiązania gwarantuje spełnienie warunku $\hat{u}(0) = 0$ bez wprowadzania składnika $\mathcal{L}_{IC}$ do funkcji kosztu.

(d) Porównaj pierwotny wynik z rozwiązaniem, w którym pierwszą warstwę ukrytą zainicjalizowano cechami Fouriera:

$$\gamma(x) = [\,\sin(2^0\pi x), \cos(2^0\pi x), \ldots, \sin(2^{L-1}\pi x), \cos(2^{L-1}\pi x)\,]\,. \tag{8}$$

Dobierz L tak, aby nie zmieniać szerokości warstwy ukrytej.

Dla każdego z powyższych przypadków stwórz następujące wykresy:

- Wykres funkcji $u(x)$, tj. dokładnego rozwiązania oraz wykres funkcji $\hat{u}(x)$, tj. rozwiązania znalezionego przez sieć neuronową
- Wykres funkcji błędu.

Stwórz także wykres funkcji kosztu w zależności od liczby epok.

Uwaga. W przypadku wykorzystania biblioteki `DeepXDE` i backendu `tensorflow` należy użyć wersji tensorflow v1.

Literatura

[1] Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. CoRR **abs/1711.10561** (2017)

[2] Lu Lu, Xuhui Meng, Zhiping Mao, George Em Karniadakis DeepXDE: A Deep Learning Library for Solving Differential Equations