

# REDES NEURAIS ARTIFICIAIS PARA A RESOLUÇÃO DA EQUAÇÃO DO CALOR EM DOMÍNIOS BIDIMENSIONAIS

<sup>a</sup>Vitória Biesek, <sup>b</sup>Anderson B. Modena, <sup>c</sup>Pedro H. A. Konzen

<sup>a,c</sup>PPGMAP, IME, Universidade Federal do Rio Grande do Sul

<sup>b</sup>IME, Universidade Federal do Rio Grande do Sul

<sup>a</sup>vitoriabiesek@gmail.com, <sup>b</sup>andersonmodena@hotmail.com, <sup>c</sup>pedro.konzen@ufrgs.br



XI ERMAC 23 - Pelotas, RS, Brasil

## Introdução

A resolução das equações de calor bidimensionais, que descrevem o fluxo de calor em corpos sólidos, desempenha um papel crucial em várias aplicações científicas e de engenharia. Este trabalho apresenta a utilização de *Physics-informed neural network* (PINN) como uma abordagem poderosa que combina métodos de aprendizado profundo com o conhecimento físico das equações diferenciais parciais (EDPs). As PINNs surgiram como uma metodologia orientada a dados para resolver EDPs, fornecendo soluções precisas e eficientes para problemas complexos. Este estudo tem como objetivo investigar as vantagens e limitações da aplicação de PINNs especificamente a equações parabólicas, com foco em equações de calor bidimensionais, contribuindo para um entendimento mais profundo de suas capacidades no contexto da modelagem de transferência de calor. Ao utilizar os conceitos de aprendizado profundo apresentados em (GOODFELLOW *et al.*, 2014) e os princípios das PINNs propostas por (RAISSI; *et al.*, 2019), esta pesquisa visa avançar no campo da engenharia térmica e fornecer insights para o desenvolvimento de soluções precisas e eficientes para problemas de transferência de calor bidimensionais.

## Problema Geral

### Equação do Calor

$$u_t = \alpha^2 \Delta u + f, \quad \mathbf{x} \in D \subset \mathbb{R}^2, \quad t > 0, \quad (1)$$

com condição de contorno

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \bar{D}, \quad (2)$$

$$u(\mathbf{x}, t) = u_0, \quad \mathbf{x} \in \partial D, \quad t > 0, \quad (3)$$

sendo  $u = u(\mathbf{x}, t)$ ,  $f = f(\mathbf{x}, t)$ .

### Rede Neural Artificial

$$\tilde{u} = \mathcal{N}(\mathbf{x}, t; \left\{ (W^{(l)}, \mathbf{b}^{(l)}, f^{(l)}) \right\}_{l=1}^{n_l}), \quad (4)$$

onde  $(W^{(l)}, \mathbf{b}^{(l)}, f^{(l)})$  é a tripla de pesos  $W^{(l)}$ , bias  $\mathbf{b}^{(l)}$  e função de ativação  $f^{(l)}$  na  $l$ -ésima camada da rede,  $l = 1, 2, \dots, n_l$ ,  $(\mathbf{x}, t) \in \bar{D} \times [0, T]$  é uma entrada da rede e  $\tilde{u} \approx u(\mathbf{x}, t)$  é a saída estimada pela rede.

### Cálculo do Resíduo

O resíduo da EDP é dado por

$$r(\mathbf{x}, t) := u_t - \alpha^2 \Delta u - f, \quad \mathbf{x} \in D, \quad t > 0, \quad (5)$$

### Treinamento da Rede

O treinamento da RNA é realizado resolvendo-se o problema de minimização

$$\min_{\{(W^{(l)}, \mathbf{b}^{(l)})\}_{l=1}^{n_l}} \left\{ \frac{1}{N_r} \sum_{i=1}^{N_r} r^2(\mathbf{x}_{r,i}, t_{r,i}) + \frac{p}{N_b} \sum_{i=1}^{N_b} [\tilde{u}(\mathbf{x}_{b,i}, t_{b,i}) - u(\mathbf{x}_{b,i}, t_{b,i})]^2 \right\}, \quad (6)$$

onde  $\{(\mathbf{x}_{r,i}, t_{r,i})\}_{i=1}^{N_r}$  são  $N_r$  pontos internos em  $D \times (0, T]$  e  $\{(\mathbf{x}_{b,i}, t_{b,i})\}_{i=1}^{N_b}$  são  $N_b$  pontos iniciais e de contorno. O parâmetro de penalização  $p$  é considerado para aumentar o peso das condições de fechamento do problema.

### Problema 0.

$$u_t = \alpha^2 \Delta u + f, \quad (t, \mathbf{x}) \in (0, T_f] \times D, \quad (7)$$

$$f = (1 + 2\alpha^2 \pi^2) e^{-T_f} \sin(\pi x_1) \sin(\pi x_2) \quad (8)$$

com condição inicial e condições de contorno dadas por

$$u(0, \mathbf{x}) = e^{-T_f} \sin(\pi x_1) \sin(\pi x_2), \quad \mathbf{x} \in D, \quad (9)$$

$$u(t, \mathbf{x}) = 0, \quad \mathbf{x} \in \partial D, \quad (10)$$

onde  $\partial D$  é a fronteira do domínio  $D = [0, 1]^2$ .

Este problema assume a solução dada por

$$u(t, \mathbf{x}) = e^{-T_f} \sin(\pi x_1) \sin(\pi x_2) \quad (11)$$

### Função Erro

A função erro é dada por

$$\begin{aligned} \varepsilon := & \frac{1}{n_s n_t} \sum_{k=1}^{n_t} \sum_{i,j=1}^{n_s} \left| \alpha^2 \Delta \tilde{u} + (1 + 2\alpha^2 \pi^2) e^{-T_f} \sin(\pi x_1) \sin(\pi x_2) - \tilde{u}_t \right|^2 \\ & + \left| \tilde{u}(0, x_1, x_2) - e^{-T_f} \sin(\pi x_1) \sin(\pi x_2) \right|^2 + \left| \tilde{u}(t, x_1, 0) \right|^2 + \left| \tilde{u}(t, 0, x_2) \right|^2 \\ & + \left| \tilde{u}(t, x_1, 1) \right|^2 + \left| \tilde{u}(t, 1, x_2) \right|^2 \end{aligned} \quad (12)$$

$$\begin{aligned} = & \frac{1}{n_s n_t} \left[ \sum_{k=1}^{n_t} \sum_{i,j=1}^{n_s} \left| \alpha^2 \left( \frac{\tilde{u}_{i+1} - 2\tilde{u}_i + \tilde{u}_{i-1}}{h^2} + \frac{\tilde{u}_{j+1} - 2\tilde{u}_j + \tilde{u}_{j-1}}{h^2} \right) \right. \right. \\ & \left. \left. + (1 + 2\alpha^2 \pi^2) e^{-T_f} \sin(\pi x_1) \sin(\pi x_2) - \frac{\tilde{u}_k - \tilde{u}_{k-1}}{h_t} \right|^2 \right. \\ & \left. + \left| \tilde{u}(0, x_1, x_2) - e^{-T_f} \sin(\pi x_1) \sin(\pi x_2) \right|^2 + \left| \tilde{u}(t, x_1, 0) \right|^2 + \left| \tilde{u}(t, 0, x_2) \right|^2 \right. \\ & \left. + \left| \tilde{u}(t, x_1, 1) \right|^2 + \left| \tilde{u}(t, 1, x_2) \right|^2 \right] \end{aligned} \quad (13)$$

## Resultados

### Versão 1. RNA $(t, x, y) \rightarrow u(t, x, y)$

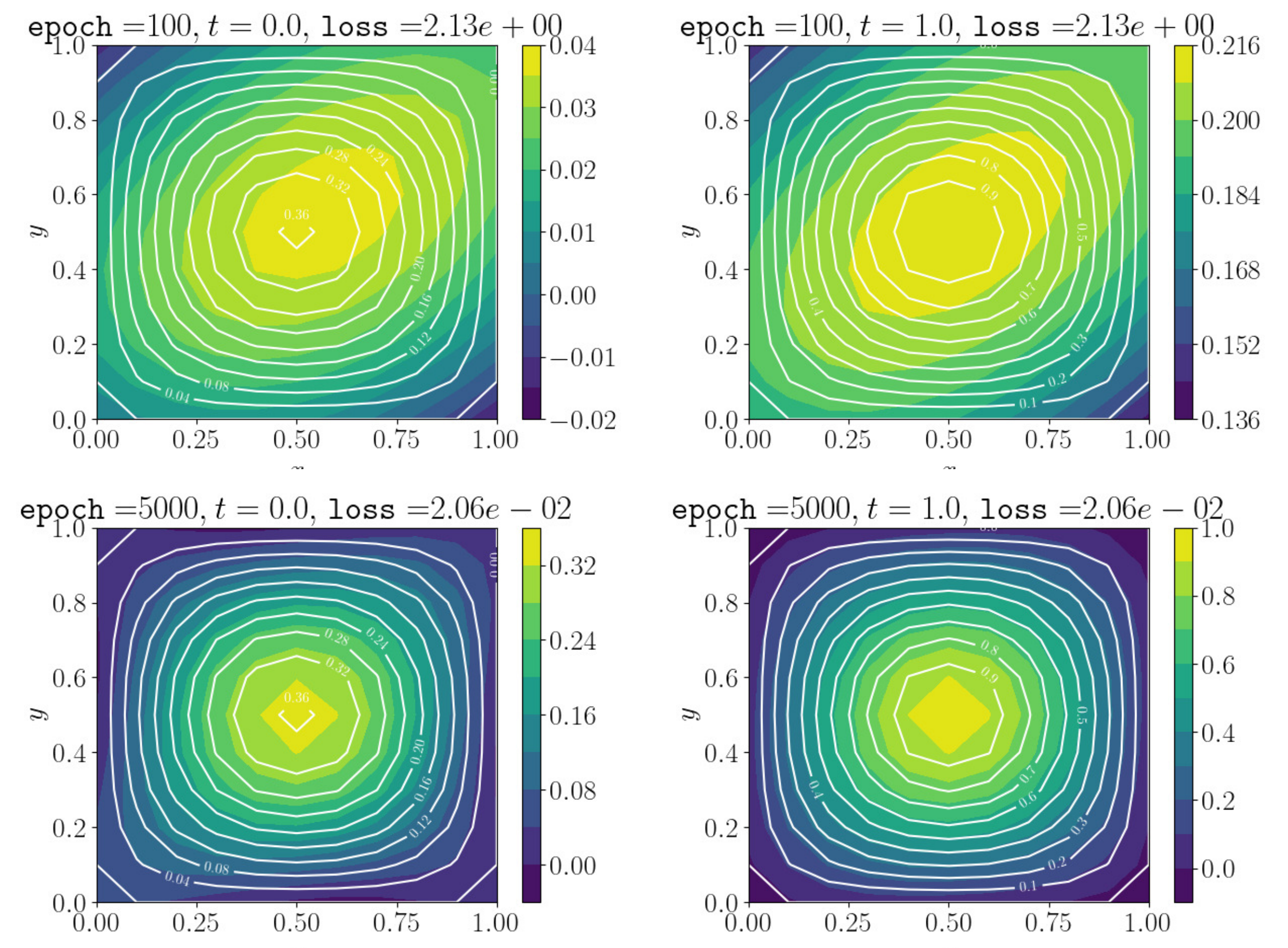


Figura 1: Estimativas da RNA na primeira e última época do treinamento para  $t = 0$  e  $t = tf$ , na primeira versão da rede.

### Versão 2. RNA $(x, y) \rightarrow u([t_0, t_1, \dots, t_n], x, y)$

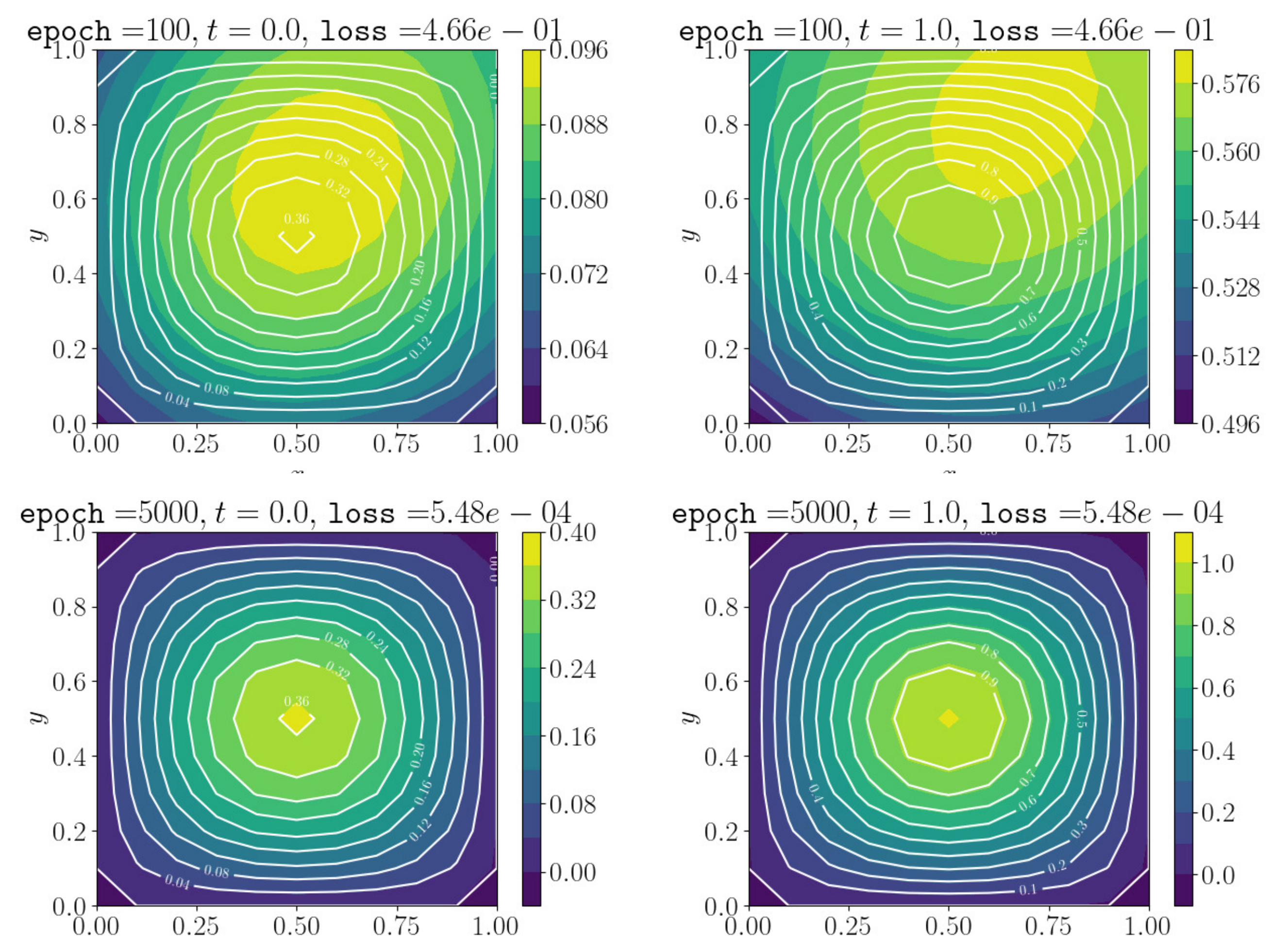


Figura 2: Estimativas da RNA na primeira e última época do treinamento para  $t = 0$  e  $t = tf$ , na segunda versão da rede.

## Considerações Finais

Os objetivos deste trabalho foram parcialmente alcançados, visto que o mesmo ainda está em andamento. Notamos que as soluções obtidas pela rede se aproximam bem da solução esperada. Como próximos passos, iremos realizar a comparação com a derivação direta da rede neural (4).

## Referências

- Goodfellow, I., Bengio, Y., Courville, A. **Deep Learning**. London: Massachusetts Institute of Technology, 2014. ISBN: 97800262035613.
- Raissi, M., Perdikaris, P., Karniadakis, G.E. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". Em: **Journal of Computational Physics** 378 (2019), pp. 686-707. DOI: 10.1016/j.jcp.2018.10.045.