HW3_Writting assignment

110652012 施品光

Lemma3.1

Statement:

Lemma 3.1. *Let $k \in \mathbb{N}_0$ and $s \in 2\mathbb{N}-1$. Then it holds that for all $\epsilon > 0$ there exists a shallow tanh neural network $\Psi_{s,\epsilon} : [-M, M] \to \mathbb{R}^{\frac{s+1}{2}}$ of width $\frac{s+1}{2}$ such that*

$$\max_{\substack{p \le s,\\ p \text{ odd}}} \left\| f_p - (\Psi_{s,\epsilon})_{\frac{p+1}{2}} \right\|_{W^{k,\infty}} \le \epsilon, \tag{17}$$

Moreover, the weights of $\Psi_{s,\epsilon}$ scale as $O\left(\epsilon^{-s/2}(2(s+2)\sqrt{2M})^{s(s+3)}\right)$ for small ϵ and large s.

Main idea:

- Polynomials are simple but essential test functions.
- This lemma proves that tanh networks can reproduce them accurately.
- The key point is that we don't need an extremely large network; the size grows only with sss.

Intuition/example:

Think of the polynomial x3x^3x3. Normally, you would draw it using its formula. Lemma 3.1 says: even without knowing the exact formula, we can set up a small tanh network that "draws" the curve of x3x^3x3 almost perfectly.

Lemma3.2

Statement:

Lemma 3.2. *Let $k \in \mathbb{N}_0$, $s \in 2\mathbb{N}-1$ and $M > 0$. For every $\epsilon > 0$, there exists a shallow tanh neural network $\psi_{s,\epsilon} : [-M, M] \to \mathbb{R}^s$ of width $\frac{3(s+1)}{2}$ such that*

$$\max_{p \leq s} \left\| f_p - (\psi_{s,\epsilon})_p \right\|_{W^{k,\infty}} \leq \epsilon. \tag{26}$$

Furthermore, the weights scale as $O\left(\epsilon^{-s/2}(\sqrt{M}(s+2))^{3s(s+3)/2}\right)$ for small ϵ and large s.

Main idea:

- Lemma 3.1 gave us one network for one polynomial.
- Lemma 3.2 gives us a more powerful result: one network can handle the whole family of polynomials up to a certain degree.

Intuition/example:

It’s like having a “universal tool” instead of many separate tools. Instead of building one network for x2x^2x2, another for x3x^3x3, etc., Lemma 3.2 guarantees there is one tanh network that can approximate all of them (up to degree sss).

Backgrounds

1. Since smooth functions can be approximated by polynomials (via Taylor’s theorem), and tanh networks can approximate polynomials, it follows that tanh networks can approximate smooth functions very well.
2. These lemmas also show that the required network size and weights grow in a controlled way, which is important for practical use.

Unanswered Questions

Could these results be extended to other activation functions, such as ReLU or sigmoid?