ML - AS3 . Written AS 111652042 顏友君

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.

Statement :

Let s be an odd positive integer ($s \in 2\mathbb{N}-1$), and $k \in \mathbb{N}_0$.

For any $\varepsilon > 0$, there exists a shallow neural network with tanh activation and width $\frac{s+1}{2}$

s.t. all odd monomials up to degree s , $f_p(y) := y^p$, $p = 1, 3, 5, \dots, s$,

can be approximated within error ε in the Sobolev norm $W^{k,\infty}([-M, M])$.

The lemma also provides an estimate for how the weights scale with ε and s $\left(O\left(\varepsilon^{-\frac{s}{2}}(2(s+2)\sqrt{2M})^{s(s+3)}\right)\right)$

Ideas :

This result takes advantage of the fact that the tanh function is odd , i.e. $\tanh(-x) = -\tanh(x)$.

Since that , shallow tanh networks are suited to represent odd functions .

Lemma 3.1 shows that with only about half as many neurons as the degree bound s ,

such a network can reproduce all odd monomials up to that degree , and also in their derivatives .

This forms the foundation for approximating polynomials by first handling the odd powers .

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*

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

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.

Statement :

Let s be an odd positive integer ($s \in 2\mathbb{N}-1$), $M > 0$, and $k \in \mathbb{N}_0$.

For any $\varepsilon > 0$, there exists a shallow tanh neural network of width $\frac{3(s+1)}{2}$

s.t. all monomials up to degree s , $f_p(y) := y^p$, $p = 1, 2, 3, \dots, s$,

can be approximated within error ε in the Sobolev norm $W^{k,\infty}([-M, M])$.

The lemma also provides an estimate for how the weights scale with ε and s $\left(O\left(\varepsilon^{-\frac{s}{2}}(\sqrt{M}(s+2))^{\frac{3s(s+3)}{2}}\right)\right)$

Ideas :

While Lemma 3.1 covers odd powers, we also require even powers.

Therefore, we need to combine multiple odd components. This increases the width of the network.

So, Lemma 3.2 expends the construction to cover both odd and even powers, at the cost of needing roughly three times as many neurons.

With this, shallow tanh networks can represent any polynomials now. :D

2. 經過觀察，我發現在MLE中的Log-likelihood function看起來很像我們Week 1所學過的Loss function，只是差了一個負號，所以它們之間有什麼特別的關係嗎？