Course: Machine Learning
Assignment: Week 2 _ Written Assignment

Student:韓韻宸(112652010)

Lemma 3.1 (Odd powers)

In simple terms:

Lemma 3.1 shows that with a shallow neural network (just one hidden layer), we can approximate functions like

$$f(x) = x, \quad x^3, \quad x^5, \quad \ldots$$

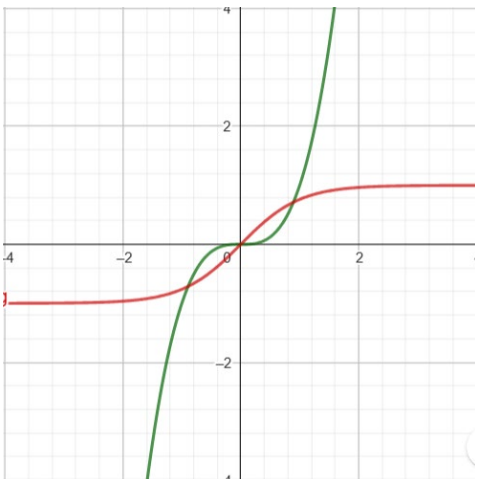
to any desired accuracy.

How does this work?

- The tanh function is an S-shaped curve.
- If we make its slope very steep (by multiplying the input with a large weight), tanh starts to look like a **step function**.
- Step functions are useful because they can mimic "switches" that turn on and off. By combining these switches, we can approximate odd polynomial shapes.

Example for Lemma 3.1 (Odd powers)

The function tanh(x) has a shape very similar to the cubic function x^3.
Both are negative for $x<0$, positive for $x>0$, and pass smoothly through the origin.
By scaling the input of tanh (making it steeper), we can make the curves almost overlap on [-1,1].



Red: tanh(x)
Green: x^3

Lemma 3.2 (Even powers)

In simple terms:
Lemma 3.2 extends the result of Lemma 3.1. It shows that we can also approximate **even powers** like

$$f(x) = x^2, \quad x^4, \quad x^6, \quad \ldots$$

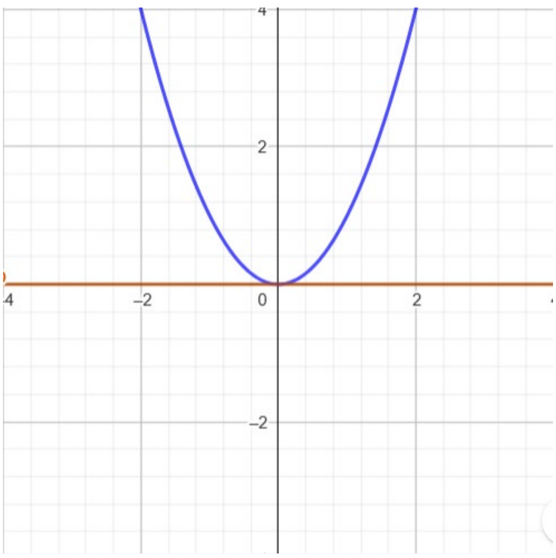
using shallow tanh networks.

How does this work?

- Odd powers were directly handled by Lemma 3.1.
- For even powers, we use a **recursive construction**: combine multiple tanh units in a clever way so that their combined effect looks like an even polynomial curve.
- This shows that both odd and even monomials are covered.

Example for Lemma 3.2 (Even powers)

The function x^2 is symmetric and U-shaped.
Although tanh itself is odd, if we combine it with its reflection, for example tanh(x)+tanh(−x), the result becomes symmetric.
This combined curve resembles the shape of x^2, especially near the origin.
By refining this construction, shallow tanh networks can approximate even-degree polynomials.



Brown: tanh(x)+tanh(-x)
Blue: x^2

Why These Lemmas Matter

- **Together, they cover all polynomials.** Odd powers (Lemma 3.1) + even powers (Lemma 3.2) = every polynomial.
- **Polynomials are dense in continuous functions.** By the Weierstrass theorem, any continuous function can be approximated by polynomials.
- **Therefore, tanh neural networks can approximate any continuous function.**

This explains why even simple neural networks with tanh activations are so powerful.

Unanswered Questions

- Why do the weights in the construction get so large when the required error ϵ is very small?

References

- Course lecture notes and assignment instructions.
- OpenAI. (2025)。ChatGPT (GPT-5) 取自https://chat.openai.com/