Unit 33: Artificial Intelligence

Universal Approximation Theorem

33.1. Multi-layer feed forward networks need to be able to model large class of functions. The basic mathematical problem has been solved already in the 1980ies in arbitrary dimensions by Cybenko and Hornik. They proved the **universal approximation theorem**. Here, we look at it in one dimensions using the **sigmoid activation function**

$$\sigma(x) = 1/(1+e^{-x}) \; .$$

A **neural network** is a finite sum of functions of the form $c\sigma(ax+b)$, where a, b, c are constants.

Every continuous $f(x)$ on $[0,1]$ can be approximated by **neural networks**.

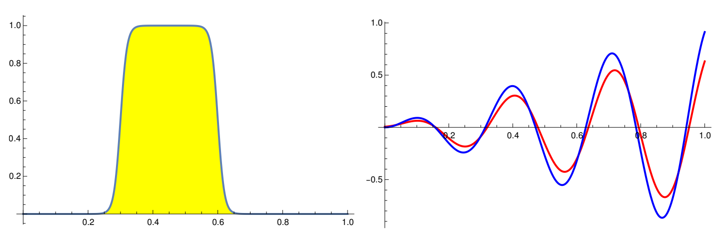

FIGURE 1. The neural network function $\sigma(100(x-0.3)) - \sigma(100(x-0.6))$ is close to the function which is 1 on $[0.3, 0.6]$ and zero else. To the right we see the function $f(x) = x\sin(x)$ and an approximation with $n = 20$ neural networks.

Proof. Pick some interval $[a, b]$ in $[0, 1]$, then look at the function

$$f(x) = \sigma(n(x-a))) - \sigma(n(x-b)) \; .$$

for large n. This function approximates the function which is 1 on $[a, b]$ and 0 else. Any function that is piece-wise constant can now be approximated by sums of such neural networks. We can therefore approximate any function. A more explicit approximation with neural networks is $g(x) = \sum_{k=1}^{n} [f(\frac{k}{n}) - f(\frac{k-1}{n})]\sigma(n(x-\frac{k}{n}))$. □

33.2. Remarks:
1) This is one of many approximation theorems. **Taylor series** allow to approximate smooth functions by polynomials. **Fourier series** allow to approximate continuous functions by sums of trigonometric functions. In artificial intelligence, the sigmoid function is nice because as we have seen, the derivative can be written again in terms of the sigmoid function. This allows fast computation of **neural networks**. In order to speed this up even more, one sometimes uses piecewise linear functions and discrete derivatives. Speed is important. Llama 3 has 70 billion parameters, GPT 4 has 1.7 trillion parameters. 2) If you should look this theorem, you will see other notation. The sum of these functions can be written as $\sigma(\vec{a}x+\vec{b})\cdot\vec{c}$, where $\vec{a},\vec{b},\vec{c}$ are vectors, arrays of numbers and the activation is applied to each entry. In higher dimensions, one uses matrices.
3) What does "approximation" means? One way to give a distance between two functions is to compute $\int_0^1 |f(x)-g(x)|\,dx$ meaning to compute the absolute area between the graphs. If this is small, then the graphs are close.

Random Functions

33.3. Let us ask an AI teacher to automatically build worksheets or exam problems as well as solutions. In order to generate problems, we first must build **random functions**. When asked "give me an example of a function", the system should generate functions of some complexity:

> **Definition:** A **basic function** is a function from the 10 functions $\{\sin,\cos,\log,\exp,\tan,\text{sqt},\text{pow},\text{inv},\text{sca},\text{tra}\}$.

33.4. Here $\text{sqt}(x)=\sqrt{x}$ and $\text{inv}(x)=1/x^k$ for a random integer k between -1 and -3, $\text{pow}(x)=x^k$ for a random integer k between 2 and 5. $\text{sca}(x)=kx$ is a scalar multiplication for a random nonzero integer k between -3 and 3 and $\text{tra}(x)=x+k$ translates for a random integer k between -4 and 4.

33.5. Second, we use addition, subtraction multiplication, division and composition to build more complicated functions:

> **Definition:** A **basic operation** is an operation from the list $\{f\circ g, f+g, f*g, f/g, f-g\}$.

33.6. The operation x^y is not included because it is equivalent to $\exp(x\log(y)) = \exp\circ(x\cdot\log)$. We can now build functions of various complexities:

> **Definition:** A **random function** of complexity n is obtained by taking n random basic functions $f_1,\dots,f_n$, and n random basic operators $\oplus_1,\dots,\oplus_n$ and forming $f_n\oplus_n f_{n-1}\oplus_{n-1}\cdots\oplus_2 f_1\oplus_1 f_0$ where $f_0(x)=x$ and where we start forming the function from the right.

Visitor: "Give me an easy function": Sofia looks for a function of complexity one: like $x\tan(x)$, or $x+\log(x)$, or $-3x^2$, or $x/(x-3)$.

Visitor: "Give me a function": Sofia returns a random function of complexity two: $x\sin(x)-\tan(x)$, or $-e^{\sqrt{x}}+\sqrt{x}$ or $x\sin(x)/\log(x)$ or $\tan(x)/x^4$.

Visitor: "Give me a difficult function": Sofia builds a random function of complexity four like $x^4 e^{-\cos(x)}\cos(x)+\tan(x)$, or $x-\sqrt{x}-e^x+\log(x)+\cos(x)$, or $(1+x)(x\cot(x)-\log(x))/x^2$, or $(-x+\sin(x+3)-3)\csc(x)$

33.7. Now, we can build a random calculus problem. To give you an idea, here are some templates for integration problems:

Definition: A **random integration problem** of complexity n is a sentence from the sentence list { "Integrate $f(x)=F(x)$", "Find the anti derivative of $F(x)$", "What is the integral of $f(x)=F(x)$?", "You know the derivative of a function is $f'(x)=F(x)$. Find $f(x)$." }, where F is a random function of complexity n.

Visitor "Give me a differentiation problem". **Sofia:** Differentiate $f(x)=x\sin(x)-\frac{1}{x^2}$. The answer is $\frac{2}{x^3}+\sin(x)+x\cos(x)$.

Visitor: "Give me a difficult integration problem". **Sofia**: Find f if $f'(x)=\frac{1}{x}+\left(3\sin^2(x)+\sin(\sin(x))\right)\cos(x)$. The answer is $\log(x)+\sin^3(x)-\cos(\sin(x))$.

Visitor: "Give me an easy extremization problem". **Sofia**: Find the extrema of $f(x)=x/\log(x)$. The answer is $x=e$.

Visitor: "Give me an extremization problem". **Sofia**: Find the maxima and minima of $f(x)=x-x^4+\log(x)$. The extrema are

$$-\frac{\sqrt{\left(9+\sqrt{3153}\right)^{2/3}-8\sqrt[3]{6}}+\sqrt{8\sqrt[3]{6}-\left(9+\sqrt{3153}\right)^{2/3}\left(1+6\sqrt{\frac{2}{9+\sqrt{3153}-8\sqrt[3]{6\left(9+\sqrt{3153}\right)}}}\right)}}{2 2^{5/6}\sqrt[3]{3}\sqrt[6]{9+\sqrt{3153}}}.$$

As we will see in class, AI content can generate homework or worksheets quickly. Automated problem generation is the "fast food" of teaching and usually not healthy.

Homework

Problem 33.1: Illustrate the universal approximation theorem to approximate the function which is 3 on $[0.1, 0.2]$ and -2 on $[0.5, 0.6]$. Include a plot of your function.

Problem 33.2: We build a differentiation problem by combining log and sin and exp. Differentiate all of the 6 combinations $\log(\sin(\exp(x)))$, $\log(\exp(\sin(x)))$,$\exp(\log(\sin(x)))$, $\exp(\sin(\log(x)))$, $\sin(\log(\exp(x)))$ and $\sin(\exp(\log(x)))$.

Problem 33.3: Four of the 6 combinations of log and sin and exp can be integrated as elementary functions.
a) Find all these cases
b) Do these integrals.

Problem 33.4: From the 10 functions f and 10 functions g and 5 operations, we can build 500 functions. Some can not be integrated. An example is $\exp(\sin(x))$. Find 4 more which can not be integrated by you now by any computer algebra system.

Problem 33.5: We are getting to the end of the course. Ask your favorite AI to write a short exam with 5 problems the entire exam should not be longer than 1 page and doable in an hour.
Problem 1. should be about continuity Problem 2. should be about limits Problem 3. should be about differentiation Problem 4. should be about integration Problem 5. Problem should be about extrema.

Oliver Knill, knill@math.harvard.edu, Math 1a, Spring, 2024