Mathematics of Image and Data Analysis
Math 5467

Universal Approximation

Instructor: Jeff Calder
Email: jcalder@umn.edu

http://www-users.math.umn.edu/~jwcalder/5467

Announcements

- Please fill out student rating of teaching (SRT) before May 2. You'll have a link in your email.
- HW2, HW3 and Project 2 have been graded. Solutions will be posted this evening.
- Final exam has been posted to the course website, due May 11.

Last Time

- Nesterov acceleration

Today

- Universal approximation

Universal approximation

Part of the success of neural networks is due to the fact that they can approximate any continuous function, given enough parameters. In particular, we can approximate any function by a **2-layer** neural network.

- This cannot fully explain this success, since other methods, like polynomial fitting, can achieve the same universal approximation results.
- It also does not explain why deeper networks can perform better, or why gradient descent finds networks that generalize.

Today we'll consider a 2-layer neural network with N hidden nodes and ReLU activations, which has the form

$$f_N(x) = \sum_{i=1}^{N} a_i(w_i x + b_i)_+, \tag{1}$$

This is a function $f_N : \mathbb{R} \to \mathbb{R}$ and the weights $a_i, w_i, b_i \in \mathbb{R}$.

Lipschitz functions

We say a function $u : \mathbb{R} \to \mathbb{R}$ is *Lipschitz continuous* if there exists $C > 0$ such that

$$|u(x) - u(y)| \leq C|x - y|. \tag{2}$$

The smallest such constant is called the *Lipschitz constant* of u and denoted

$$\mathrm{Lip}(u) = \sup_{\substack{x,y \in \mathbb{R} \\ x \neq y}} \frac{|u(x) - u(y)|}{|x - y|}. \tag{3}$$

Universal approximation

Theorem 1. *Let $\varepsilon > 0$, let $u : \mathbb{R} \to \mathbb{R}$ be Lipschitz continuous, and let $R > 0$. There exists a 2-layer ReLU neural network $f_N(x)$ of the form (1) with $N = 6(R\,\mathrm{Lip}(u)\varepsilon^{-1} + 1)$ hidden nodes such that*

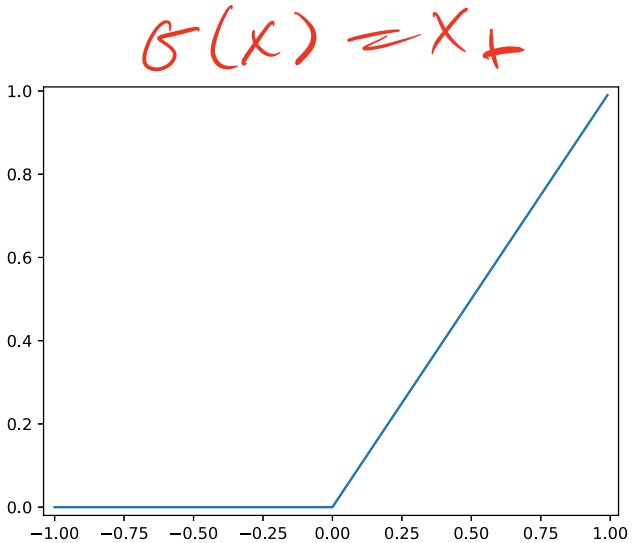
$$\max_{-R \le x \le R} |f_n(x) - u(x)| \le \varepsilon. \tag{4}$$

Furthermore, if u' is Lipschitz continuous then we need only

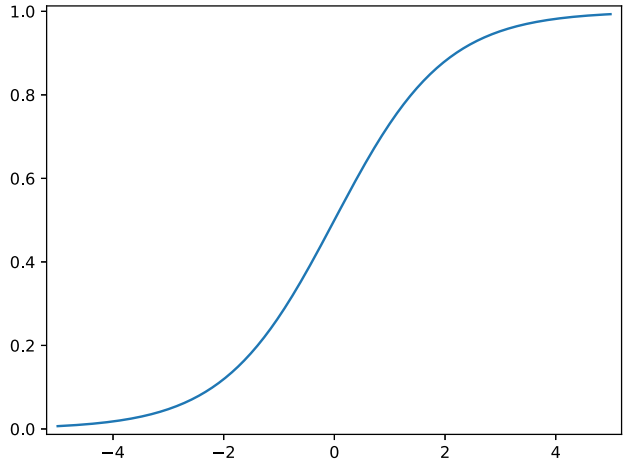
$$N = 6(R\sqrt{\mathrm{Lip}(u')\varepsilon^{-1}} + 1) \tag{5}$$

hidden nodes.

Activations



(a) ReLU



(b) Sigmoid

Figure 1: Plots of the ReLU and Sigmoid activation functions. Both activation functions have the behavior that they give zero, or close to zero, responses when the input is below a certain threshold, and give positive responses above.

Bump functions

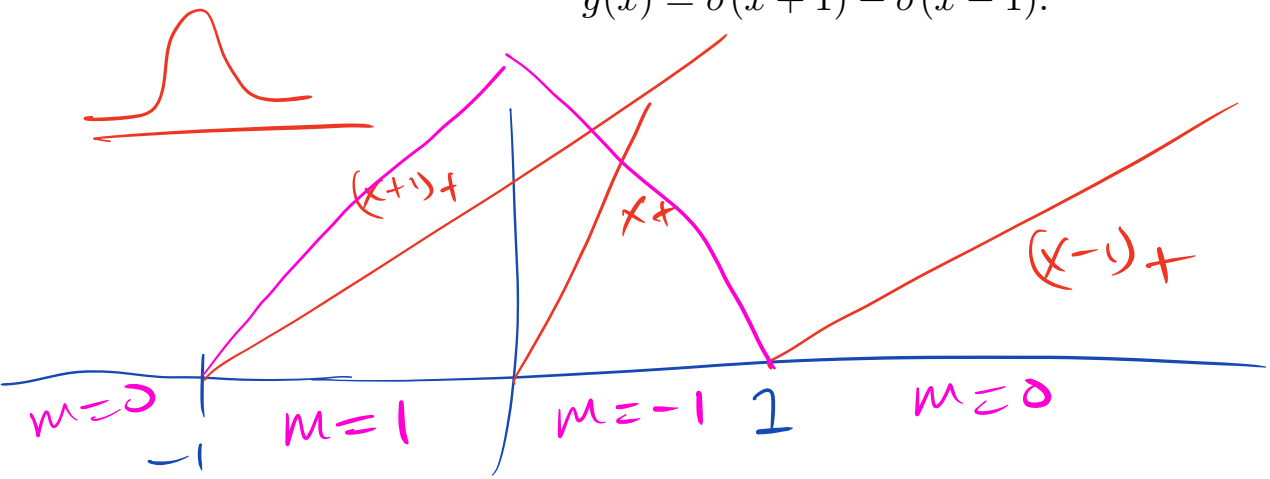
The proof is based on the construction of bump functions out of 2-layer networks. For ReLU the bump function is

$$g(x) = (x+1)_+ - 2x_+ + (x-1)_+.$$

3 neurons

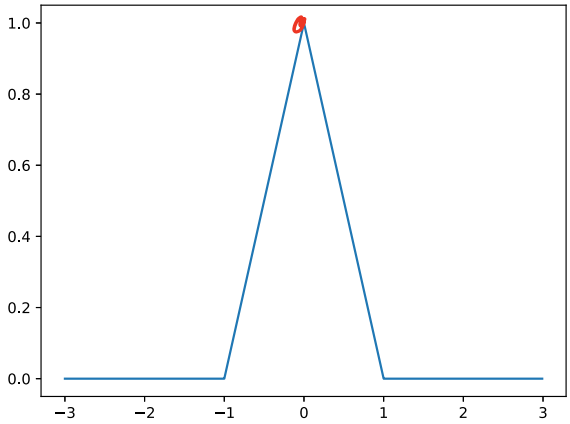
For other activations the construction can be different. For the sigmoid activation a bump function is

$$g(x) = \sigma(x+1) - \sigma(x-1).$$

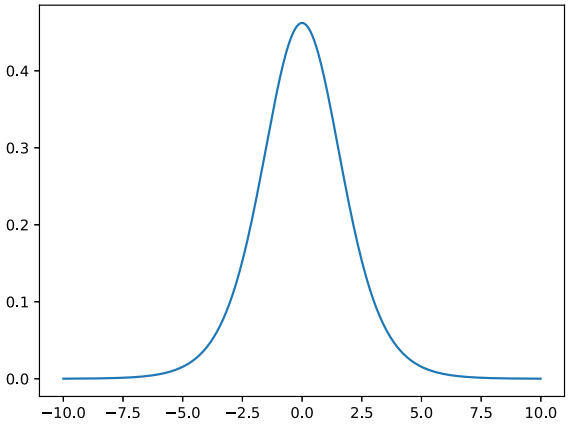


Bump functions

$g(x) = 0$ if $|x| \geq 1$



(a) ReLU bump



(b) Sigmoid bump

Figure 2: Examples of bump functions constructed from 2-layer neural networks with ReLU and sigmoid activations.

Proof of Universal Approximation Theorem

ReLU Bump: $g(x) = (x+1)_+ - 2x_+ + (x-1)_+$

$$a g(b(x-c)) = a g(bx - bc)$$

$$= a(bx - bc + 1)_+ - 2a(bx - bc)_+ + a(bx - bc - 1)_+$$

2-layer ReLU network.

A 2-layer ReLU can implement

$$\sum_{i=1}^{N} a_i g(b_i(x-c_i))$$

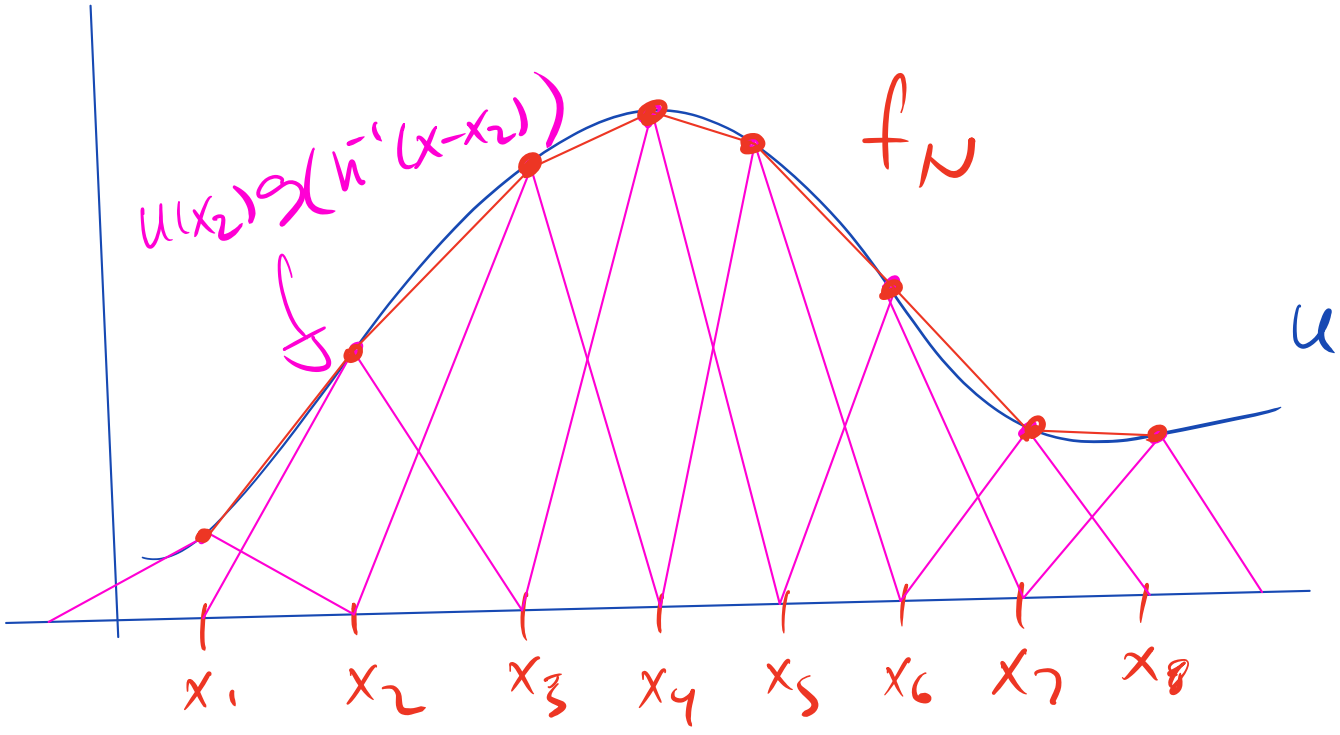
Let $x_i = hi$, $i \in \mathbb{Z}$, $m = \lceil \frac{1}{h} \rceil$

Defin: $f_N(x) = \sum_{i=-m}^{m} u(x_i) g(h^{-1}(x-x_i))$

Claim: $f_N(x_j) = u(x_j)$

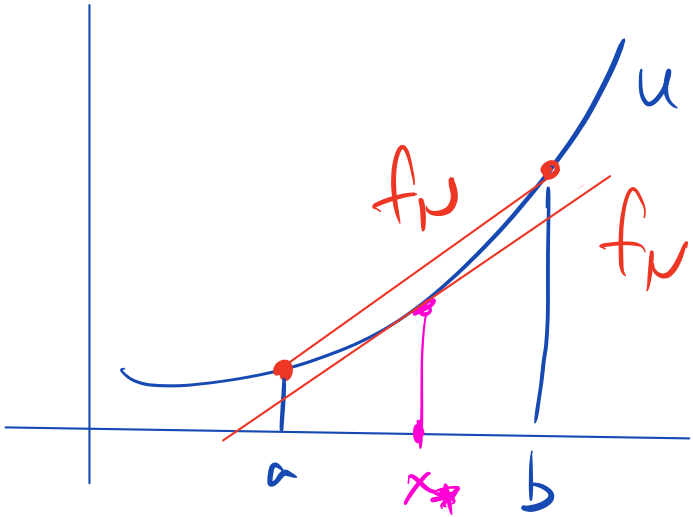
$$g(h^{-1}(x_j - x_i)) = \begin{cases} 1, & i=j \\ 0, & i \neq j \end{cases}$$

$|x_i - x_j| \geq h$ if $i \neq j$



f_N = piecewise linear + continuous interpolation of u at points x_i.

How close is f_N to u?



$$f_N(x) = u(a) + (x-a)\frac{u(b)-u(a)}{b-a}$$

$$u'(x_*) = m = \frac{u(b)-u(a)}{b-a}$$

Mean value theorem

① $|f_N(x) - u(x)|$

$$= \left| u(a) + \frac{(x-a)}{b-a}(u(b)-u(a)) - u(x) \right|$$

$$= \left| u(a) - u(x) + \frac{(x-a)}{b-a}\,(u(b)-u(a)) \right|$$

$$\le |u(a)-u(x)| + \underbrace{\frac{|x-a|}{b-a}}_{\le 1}\,|u(b)-u(a)|$$

$$\le \mathrm{Lip}(u)\,|b-a| + \mathrm{Lip}(u)\,|b-a|$$

$$= 2\,\mathrm{Lip}(u)\,|b-a|.$$

$$b-a = x_{i+1} - x_i = h \le \varepsilon$$

So $|f_N - u| \le 2\,\mathrm{Lip}(u)\,h.$

Number of nodes $= O(h^{-1})$

Thm $O(h^{-1}) \geq O(\varepsilon^{-1})$

If u' is Lipschitz, write

$$f_N(x) \leq m(x-a) + u(a)$$

$$m = \frac{u(b)-u(a)}{b-a}$$

MVT $\Rightarrow \exists x_* \in [a,b]$

such that $u'(x_*) = m$

$$|f_N(x) - u(x)| = |m(x-a) + u(a) - u(x)|$$

$m = u'(x_*)$

$$= |m(x - x_*) + m(x_* - a) + u(a) - u(x)|$$

$$= |u(a) - u(x_*) - u'(x_*)(a - x_*) + u(x_*) + u'(x_*)(x - x_*) - u(x)|$$

$$\leq |u(a) - u(x_*) - u'(x_*)(a - x_*)| + |u(x_*) + u'(x_*)(x - x_*) - u(x)|$$

$$\leq \frac{1}{2}\,\mathrm{Lip}(u')\left(|a-x_*|^2+|x-x_*|^2\right)$$

$$\leq \mathrm{Lip}(u')\,|b-a|^2 \qquad \leq 2\,|b-a|^2$$

$$\leq \varepsilon$$

$$|f_N - u| \leq \mathrm{Lip}(u')\,h^2 \leq \varepsilon, \quad b-a=h$$

$$N = O(h^{-1}), \quad h^{-1} = O(\varepsilon^{-1/2})$$

Overfitting

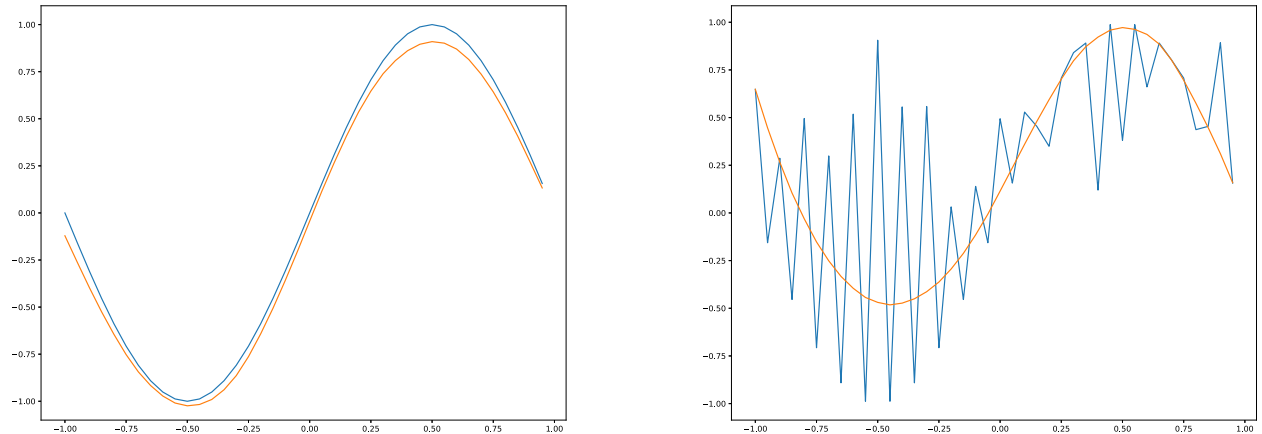
The function

$$f_N(x) = \sum_{i=-m}^{m} y_i g(h^{-1}(x - x_i)),$$

constructed in the proof exactly fits the data $f_N(x_i) = y_i$. This means a network with m nodes can exactly fit m datapoints, simply via memorization.

In practice it is hard (but not impossible) to get neural networks to overfit like this, even in the severely over parameterized regime. This is somewhat of a mystery still in the community, but is related to the optimization algorithms used to train neural networks.

Noisy labels



(a) Smooth labels

(b) Noisy labels

Figure 3: Example of a network fitting smooth and noisy data. The network has 10000 hidden nodes and only 40 training points.

Approximating polynomials with deep ReLU networks

Theorem 2. *Let g be a polynomial of degree k. For any $\varepsilon > 0$ there exists a ReLU network f with $O(k \log(\varepsilon^{-1}))$ hidden nodes such that*

$$|f(x) - g(x)| \leq \varepsilon \quad \textit{for } 0 \leq x \leq 1.$$

Approximating x^2 deep ReLU networks

Theorem 3. *For any $\varepsilon > 0$ there exists a ReLU network f with $O(\log(\varepsilon^{-1}))$ layers and nodes such that*

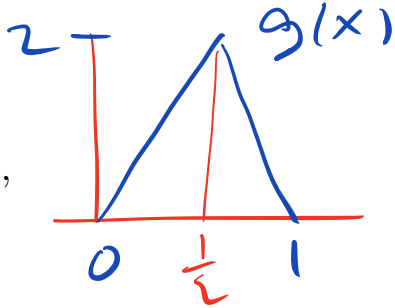
$$|f(x) - x^2| \leq \varepsilon \quad \textit{for } 0 \leq x \leq 1.$$

$\varepsilon = 10^{-9}$ $\log(\varepsilon^{-1}) = 9$

Sawtooth functions

We define

$$g(x) = \begin{cases} 2x, & \text{if } 0 \le x \le \frac{1}{2} \\ 2 - 2x, & \text{, if } \frac{1}{2} \le x \le 1, \end{cases}$$

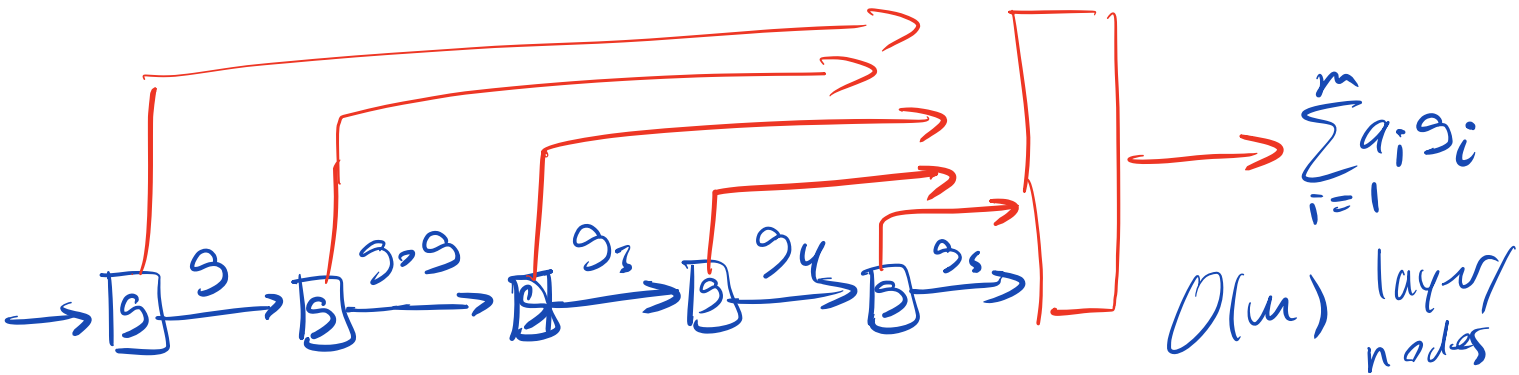


and the m-fold composition

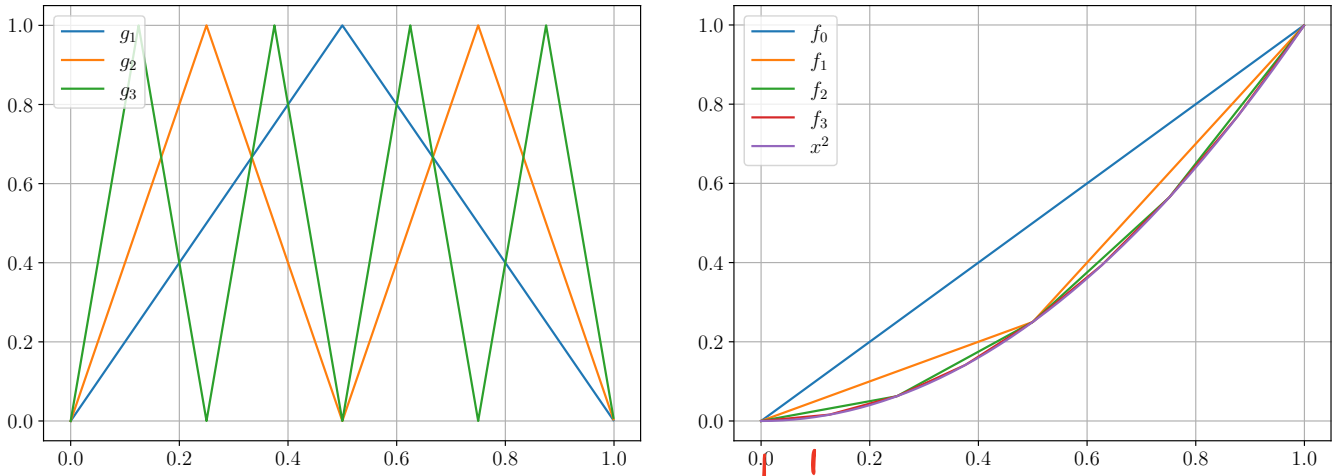
$$g_m = \underbrace{g \circ g \circ \cdots \circ g}_{m \text{ times}}.$$

Also let f_m be the piecewise linear approximation of x^2 with 2^m pieces on $[0, 1]$. Then as before

$$|f_m(x) - x^2| \le 2^{-2m}.$$



Sawtooth functions



(a) g_i (b) f_i

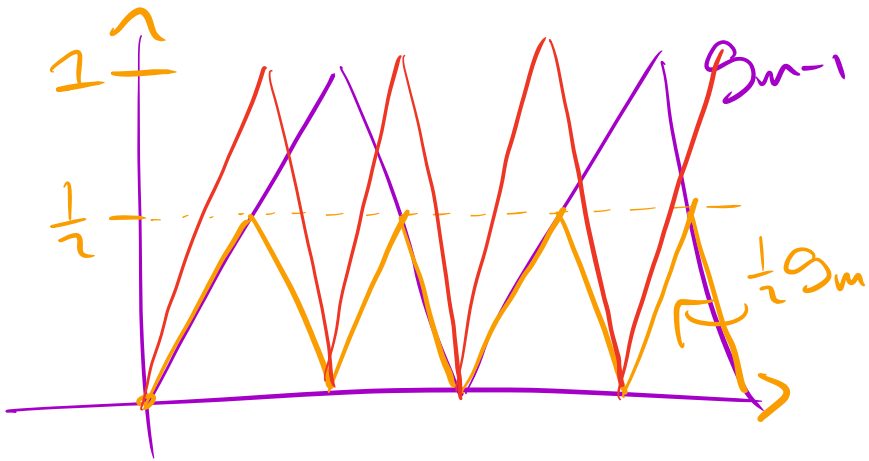
Figure 4: A depicition of the functions f_i and g_i used in approximating the parabola x^2.

Sawtooth functions

$$g_m(x) = g(g_{m-1}(x))$$

$$g_m'(x) = g'(g_{m-1}(x))\, g_{m-1}'(x)$$

$$= \begin{cases} 2\, g_{m-1}'(x), & \text{if } g_{m-1}(x) \le \frac{1}{2} \\ -2\, g_{m-1}'(x), & \text{if } g_{m-1}(x) \ge \frac{1}{2} \end{cases}$$



Key lemma

Lemma 4. *For any $0 \le x \le 1$ and $m \ge 1$ we have*

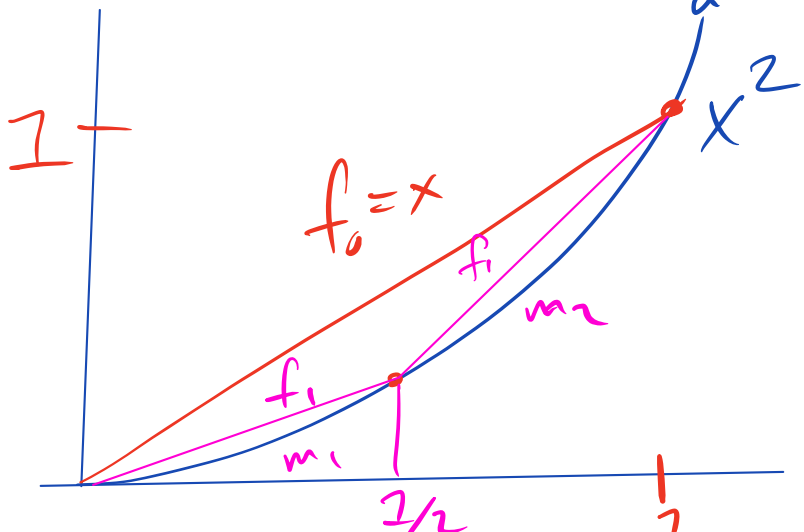
$$f_m(x) = x - \sum_{k=1}^{m} \frac{g_k(x)}{2^{2k}}. \tag{6}$$

O(m) layers + nodes

$|f_m - x^2| \le 2^{-2m} = \varepsilon$

$m = O(\log(\varepsilon^{-1}))$

Claim $f_m - f_{m-1} = \frac{-g_m}{2^{2m}}$



$f_0' = 1$

$m_1 = \frac{(\frac{1}{2})^2 - 0}{\frac{1}{2}}$

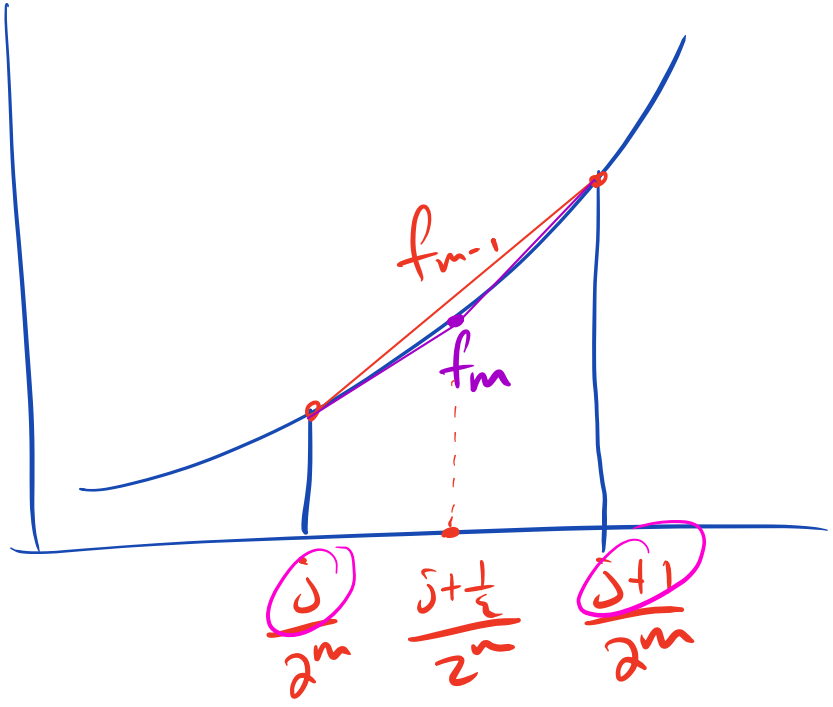
$m_1 = \frac{1}{2} = f_0' - \frac{1}{2}$

$m_2 = \frac{1-(\frac{1}{2})^2}{\frac{1}{2}}$

$= 2 - \frac{1}{2} = \frac{3}{2} = f_0' + \frac{1}{2}$

$$f_1'(x) = \begin{cases} f_0'(x) - \frac{1}{2}, & 0 < x \le \frac{1}{2} \\ f_0'(x) + \frac{1}{2}, & \frac{1}{2} \le x \le 1 \end{cases}$$

$$f_1'(x) - f_0'(x) = \frac{-g_1'(x)}{4} = 2^{2m}, \; m = 1$$



$$f'_{m-1}(x) = \frac{\left(\frac{j+1}{2^m}\right)^2 - \left(\frac{j}{2^m}\right)^2}{\frac{1}{2^m}} = \dots$$

$$f_m^1 - f_{m-1}^{(} = \pm \frac{1}{2^m}$$

Polynomials

$$xy = \frac{1}{4}(x+y)^2 - \frac{1}{4}(x-y)^2$$

↳ $x^3, x^4 \longrightarrow$ polynomials