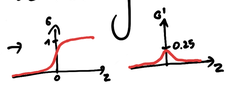
Vanishing Gradients:

```
class MyNN:
    def __init__(self):
        self.Θ_1 = ...
        self.Θ_2 = ...
        self.Θ_3 = ...
    def forward(x):
        x = relu(x @ Θ_1)
        x = relu(x @ Θ_2)
        if classification:
            x = sigmoid(x @ Θ_3)
        else:
            x = x @ Θ_3
        return x
```

good initialization

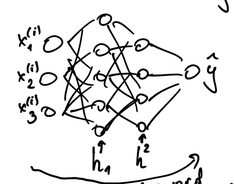
$\sigma(z) = \frac{1}{1+e^{-z}} \rightarrow$



$x^{(i)} \in \mathbb{R}^3 \quad h_1 \in \mathbb{R}^5 \quad h_2 \in \mathbb{R}^4$

Classification $\hat{y} \in [0, 1]$

Regression $\hat{y} \in \mathbb{R}$



$h_1 = \sigma(x^{(i)} \Theta^{(1)}) \qquad g_1 = x^{(i)} \Theta^{(1)}$

$h_2 = \sigma(h_1 \Theta^{(2)}) \qquad g_2 = h_1 \Theta^{(2)}$

Classification: $\hat{y} = \sigma(h_2 \Theta^{(3)}) \in [0, 1]$

Regression: $\hat{y} = h_2 \Theta^{(3)} \in (-\infty; +\infty)$

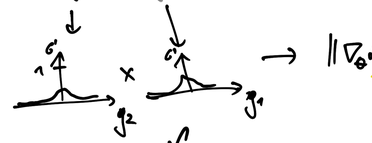
still sigmoid, because $\hat{y}$ needs $[0, 1]$

$\ell = y_i - \hat{y}_i$

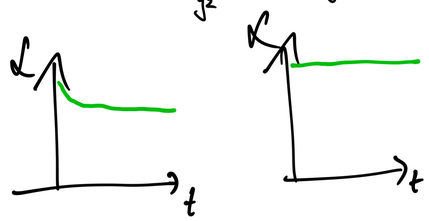
$\mathcal{L} = \sum_i (y_i - \hat{y}_i)^2 = \sum_i \ell^2$

$\nabla_\Theta \mathcal{L} = \frac{\partial \mathcal{L}}{\partial \ell} \cdot \frac{\partial \ell}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial \Theta} =$

$= \sum_i^{\mathcal{L}} 2\ell \cdot \frac{\partial \ell}{\partial h_2} \underbrace{\frac{\partial h_2}{\partial g_2}}_{\sigma'(g_2) \approx 0} \frac{\partial g_2}{\partial h_1} \underbrace{\frac{\partial h_1}{\partial g_1}}_{\sigma'(g_1) \approx 0} \frac{\partial g_1}{\partial \Theta} =$

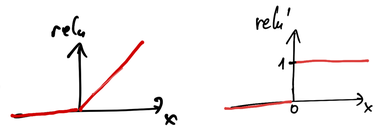


$\rightarrow \|\nabla_\Theta \mathcal{L}\| \ll 1$ (≈ 0)



Exploding Gradients:

$\text{relu}(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases} \longrightarrow$



$\hat{y} = \text{relu}(\text{relu}(x^{(i)} \Theta^{(1)}) \Theta^{(2)}) \Theta^{(3)}$

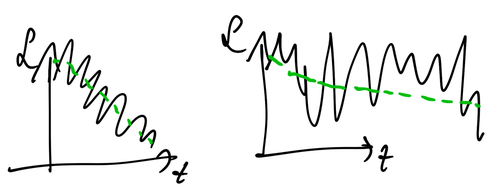
$\nabla_\Theta \mathcal{L} = \frac{\partial \mathcal{L}}{\partial h_2} \underbrace{\frac{\partial h_2}{\partial g_2}}_{\begin{cases} 0 & \text{if } g_2 < 0 \\ 1 & \text{if } g_2 \geq 0 \end{cases}} \frac{\partial g_2}{\partial h_1} \underbrace{\frac{\partial h_1}{\partial g_1}}_{\begin{cases} 0 & g_1 < 0 \\ 1 & g_1 \geq 0 \end{cases}} \frac{\partial g_1}{\partial \Theta}$

$\sum_i^B \ldots$

if $\left\|\frac{\partial g_2}{\partial h_2}\right\| \gg 1$ & $\left\|\frac{\partial g_1}{\partial \Theta}\right\| \gg 1$ ≈ 10

$\Downarrow$

$\|\nabla_\Theta \mathcal{L}\| \gg 1 \approx 10^M$ M-num layers



Solutions:

- Normalize inputs ($x \in [0.2, 0.5] \dashrightarrow [-1, 1]$)
- Choose appropriate activ. fns. (relu?)
- Better learning algorithm (adaptive) - ADAM, Adagrad, etc.
- Weight initialization ($\Theta_{t=0} \sim \mathcal{N}(0, \frac{1}{\sqrt{M}})$, M - num weights)

Solutions:

- Normalize inputs ($x \in [0, 125000] \dashrightarrow [-1, 1]$)
- Gradient clipping ($\nabla_\Theta \mathcal{L} := \text{clip}(\nabla_\Theta \mathcal{L}, -10, +10)$)
- Smaller batch-size ($B = 128 \dashrightarrow B = 16$)
- Regularization ($\mathcal{L} += \lambda \sum_j^M \Theta_j^2$)
- Better learning algorithm (adaptive) - ADAM, Adagrad, etc.
- Weight initialization ($\Theta_{t=0} \sim \mathcal{N}(0, \frac{1}{\sqrt{M}}) \ldots$)