Lecture 7: Impact of the Architecture

Soufiane Hayou

Wednesday 31st May, 2023

Choice of Activation function

If $|\sigma'(x)| < 1$ for all x, we might have a gradient vanishing problem

Choice of Activation function

If $|\sigma'(x)| < 1$ for all x, we might have a gradient vanishing problem
Example in a simple case $d_1 = d_2 = \cdots = d_L = 1$,

$$p_t = w_t \sigma'(w_t x_t) p_{t+1}, \qquad p_{T+1} = \nabla_{x_{T+1}} \ell(x_{T+1}, W_T).$$

$\rightarrow$ This can lead to gradient vanishing (GV)/ gradient exploding (GE) in the limit of large depth

1. $\sigma(x) = \max(x, 0), \quad \sigma'(x) = 1_{x>0}$ (no GV)
2. $\sigma(x) = \frac{1}{1+e^{-x}}, \quad \sigma'(x) = \frac{e^{-x}}{(1+e^{-x})^2}$ (potential GV)

$\rightarrow$ Activations with $|\sigma'(x)| > 1$ are not used in practice as it might lead to numerical instability (gradient exploding)

Choice of the Initialization weights

Can $|w_t|$ be initialized to avoid a gradient vanishing problem (only at initialization)?

$$r_t = \frac{p_t}{p_{t+1}} = w_t \sigma'(w_t x_t),$$

$\rightarrow$ We would like to control the growth rate $\frac{p_t}{p_{t+1}}$. Assume that σ is ReLU and $w_t \sim \mathcal{N}(0, \gamma^2)$. What is a good choice for γ?

1. $\mathbb{E}_W[r_t] = 0$
2. $\mathbb{E}_W[r_t^2 \mid x_t > 0] = \frac{\gamma^2}{2}$

$\rightarrow$Hence, we should choose $\gamma^2 = 2$. What about general widths $d_1, d_2, \ldots, d_L$?

In DNNs with width d, this becomes $\gamma_d^2 = 2/d$. This is known as Kaiming (or He) initialization scheme. More generally, with different widths d_t, we have

$$W_t^{ij} \sim \mathcal{N}(0, \frac{2}{d_t}).$$

$\rightarrow$ This also solves a problem of vanishing/exploding during forward propagation!!

Can we solve GV by changing the architecture?

Residual Neural Networks (ResNet)

Just like FC-DNN and CNN, a depth T residual neural network consists of set of T layers stacked one after another. However, ResNet learns the 'residuals' between two consecutive layers instead of the learning directly one layer from the previous one.

$$x_{t+1} = x_t + \mathcal{F}_t(x_t), \qquad t = 0, \dots, T-1 \tag{1}$$

where $\mathcal{F}_t$ is a mapping that defines the t^{th} 'residual block'.

Residual Neural Networks (ResNet)

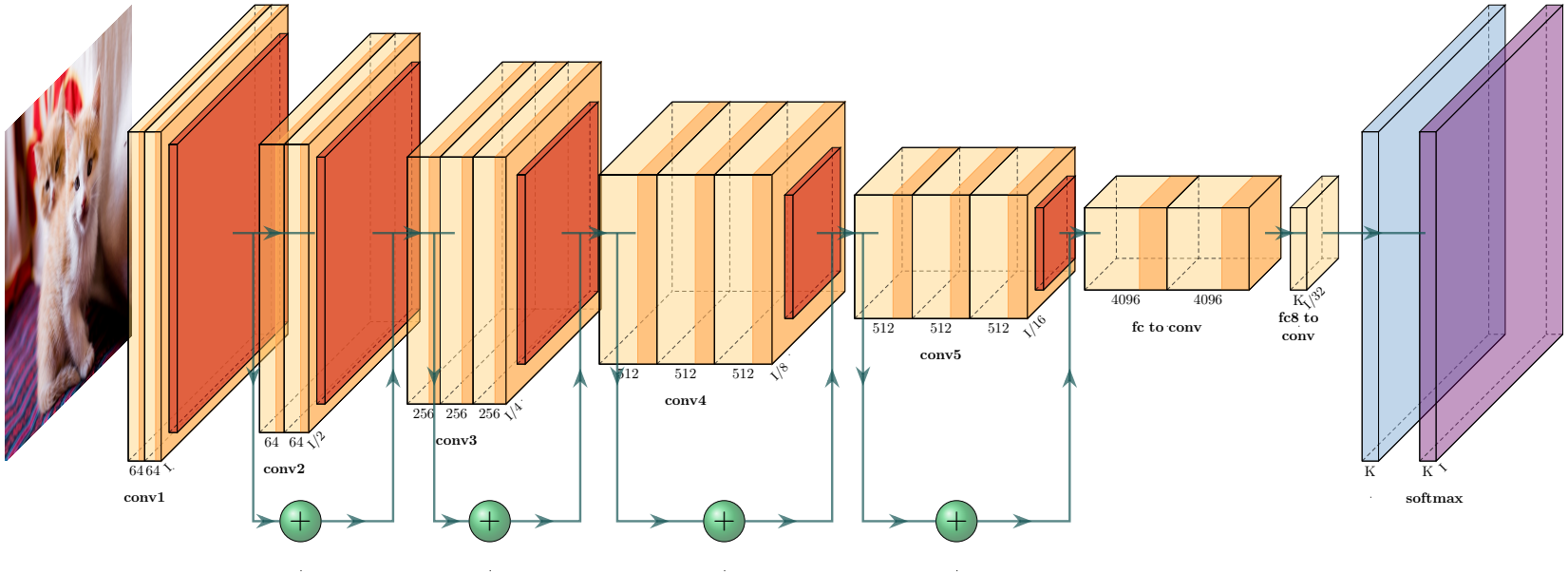

Figure: Example of a ResNet architecture.

Residual Neural Networks (ResNet)

The ResNet architecture was first introduced as an *engineered* solution to the GV problem. Observe that in the simple case with 1 neuron per layer, we have

$$\frac{\partial \ell}{\partial x_t} = \left[\prod_{k=t}^{T-1}\left(1 + \frac{\partial \mathcal{F}_k(x_k)}{\partial x_k}\right)\right] \times \frac{\partial \ell}{\partial x_T}$$

$\rightarrow$ The additional term 1 plays an crucial role in preserving the magnitude of the gradient as it back-propagates though the network.

$\rightarrow$ This can however lead to gradient exploding!! we can mitigate this issue by scaling the blocks $\mathcal{F}_k$ with constants α_k.

Scaled Blocks

Simple ResNet

Consider a simple ResNet of width d, with ReLU activation, given by

$$\begin{aligned} x_0 &= W_{in}x \\ x_{t+1} &= x_t + W_t\sigma(x_t), \qquad t = 0, \dots, T-1 \\ f(x) &= T_{\text{fc}}x_T, \end{aligned} \tag{2}$$

with $W_t^{ij} \sim \mathcal{N}(0, \frac{\gamma^2}{d})$

$\rightarrow$ The distribution of x_t changes with t. More precisely, at initialization, we have that $\mathbb{E}[(x_t^i)^2] = \Theta\left(\left(1 + \frac{\gamma^2}{2}\right)^t\right)$.

Consider a simple ResNet of width d, with ReLU activation, given by

$$\begin{aligned} x_0 &= W_{in}x \\ x_{t+1} &= x_t + W_t\sigma(x_t), \qquad t = 0, \dots, T-1 \\ f(x) &= T_{\text{fc}}x_T, \end{aligned} \tag{2}$$

with $W_t^{ij} \sim \mathcal{N}(0, \frac{\gamma^2}{d})$

$\rightarrow$ The distribution of x_t changes with t. More precisely, at initialization, we have that $\mathbb{E}[(x_t^i)^2] = \Theta\left(\left(1 + \frac{\gamma^2}{2}\right)^t\right)$.

$\rightarrow$ Can we solve this using some kind of scaling?

Demo