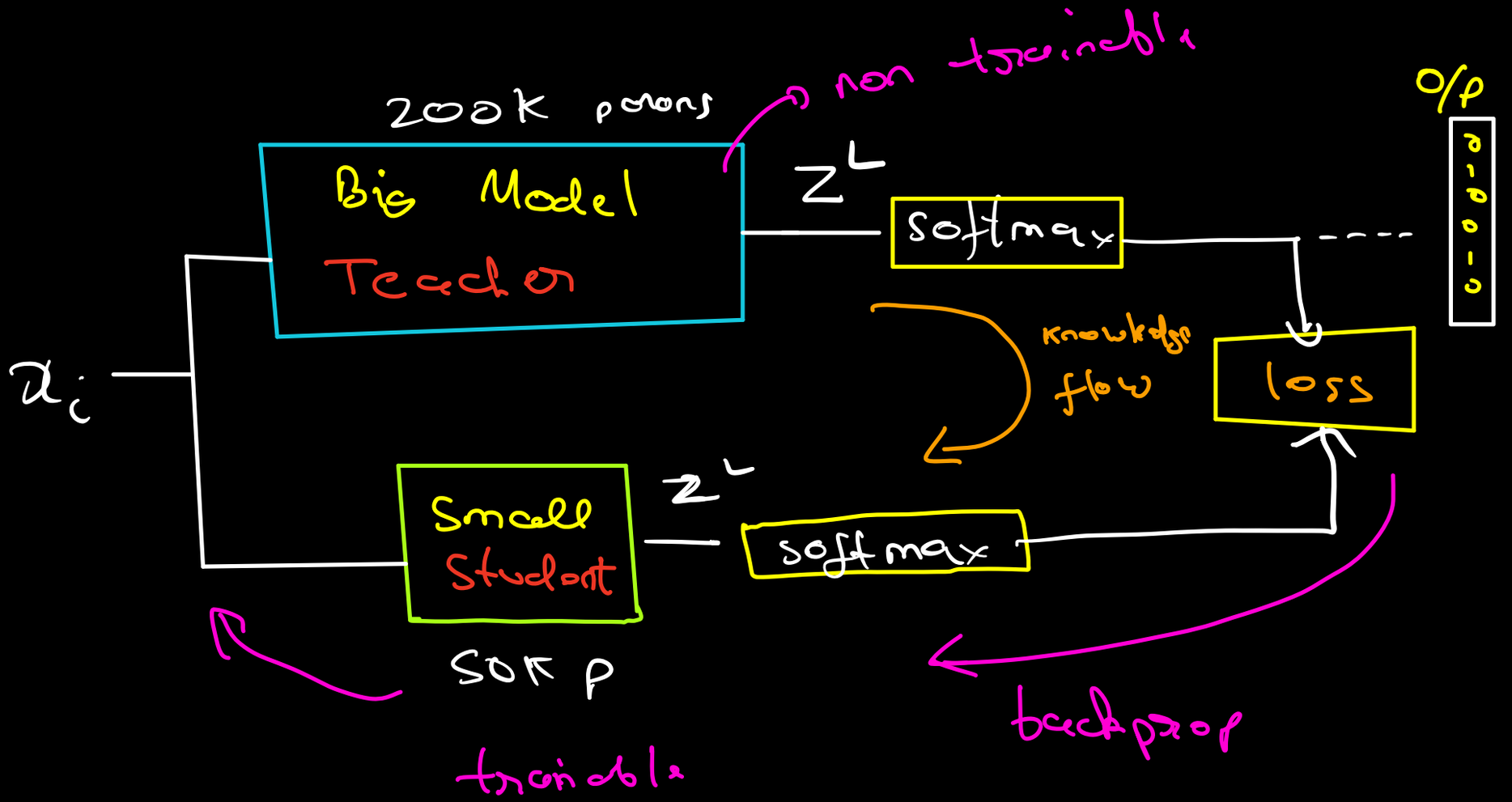
Knowledge Distillation

(2015) → paper

Issues with DL models

→ Most modern models are very large
→ There is high latency / prediction time
→ Cost of running them is high

Distillation refers to the transfer of information from a big model to a small one.



Knowledge can flow from larger model to smaller model

Which loss?

→ Loss func^n can be chosen depend^g on the application. However this setup is for classificat^n

→ After softmax, we get probabilty.

Q: What is a way to compare prob. dist?

→ KL divergence loss

$$\sum_i p_i \log\left(\frac{p_i}{q_i}\right)$$

Temperature Controlled Softmax

For distillation, in above setup we have "softmax". One issue with it is:

Eg:

$z_1 = 20$

$z_2 = 40$

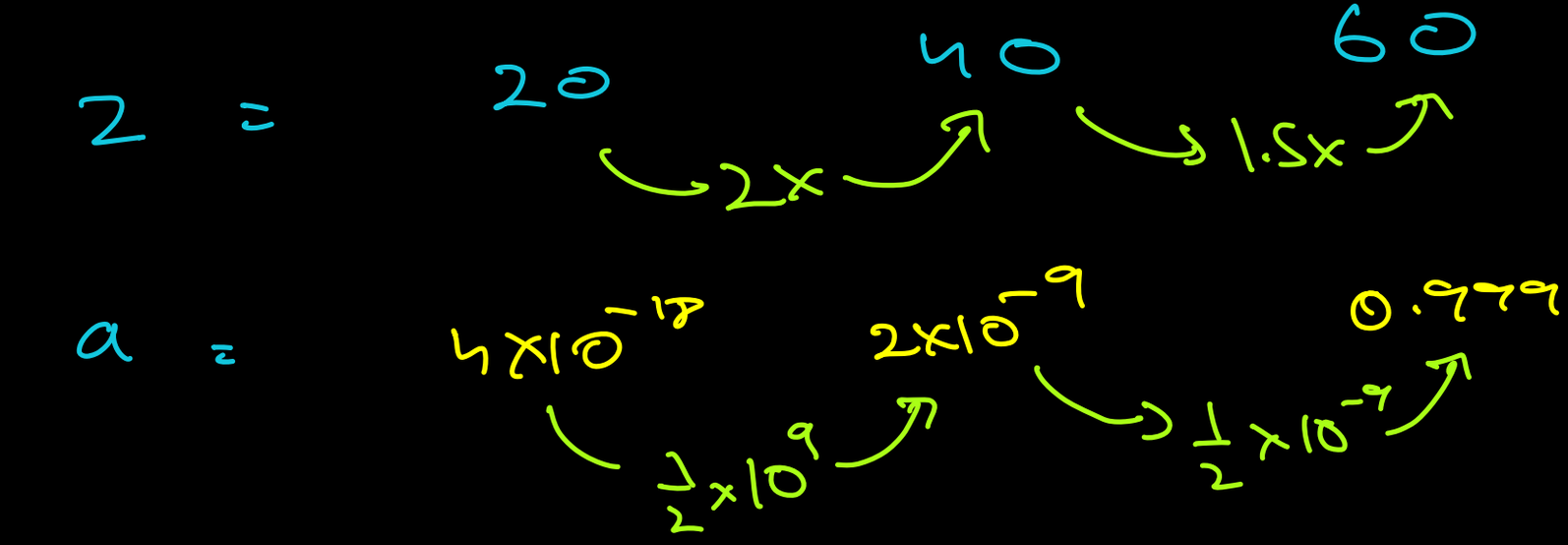
$z_3 = 60$

$$P(1) = \frac{e^{20}}{e^{20} + e^{40} + e^{60}} = 4 \times 10^{-18}$$

$$P(2) = \frac{e^{40}}{\Sigma} = 2 \times 10^{-9}$$

$$P(3) = \frac{e^{60}}{\Sigma} = 0.999\ldots$$

Squashing effect is too large!!



So after softmax, most of the time, o/p looks like this →

0.000...	0.00...	0.999...	0.000
≈ 0	0	1	0

So softmax erases a lot of learning by squashing!

Temperature -> $S_i = \frac{e^{z_i/T}}{\sum e^{z_i/T}}$

'T' is just a factor. (scaling)

eg: T = 10

$$\frac{e^{20/10}}{\Sigma}, \frac{e^{40/10}}{\Sigma}, \frac{e^{60/10}}{\Sigma}$$

$$= 0.0158, 0.1173, 0.8668$$

↑ more usable

Practical Setup