10/21

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \end{bmatrix} \begin{bmatrix} x_1 & 0 \\ x_2 & x_5 \\ 0 & x_6 \end{bmatrix}$$

sparse activation

$$\begin{bmatrix} w_1 & 0 & w_3 \\ 0 & w_5 & w_6 \end{bmatrix} \begin{bmatrix} x_1 & x_4 \\ x_2 & x_5 \\ x_3 & x_6 \end{bmatrix}$$

sparse weight

↳ activation sparsity = sparsity the activation of NN.

Dense = fully connected

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \end{bmatrix} \begin{bmatrix} x_1 & x_4 \\ x_2 & x_5 \\ x_3 & x_6 \end{bmatrix}$$

ReLU activation : $\sigma(z) = \max\{z, 0\}$

If z is a weighted sum of many zero-mean i.i.d. values, CLT tells you that $z \sim N(0, \sigma^2)$

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-u)^2}{2\sigma^2}}$$

↳ 문제점 : 2018 「Rhu et al」 show that, as we train the model, the activation density evolves by the first dropping sharply, and gradually growing.

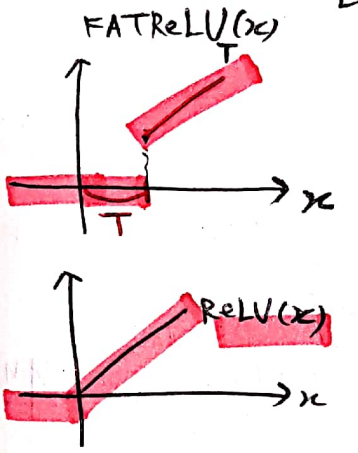
<1>

---

① l_1 regularization : $L_1(\vec{z}) = \|z\|_1$

② Hoyer regularization : $H(\vec{z}) = \frac{\|\vec{z}\|_1^2}{\|\vec{z}\|_2^2}$ 「regularize the activations explicitly」

FATReLU$_T(x)$ [x, when $x \geq T$ / 0, elsewhere.



* Nevertheless, writing kernels well gives you some speedup.

Top-k transformers are known to have some.

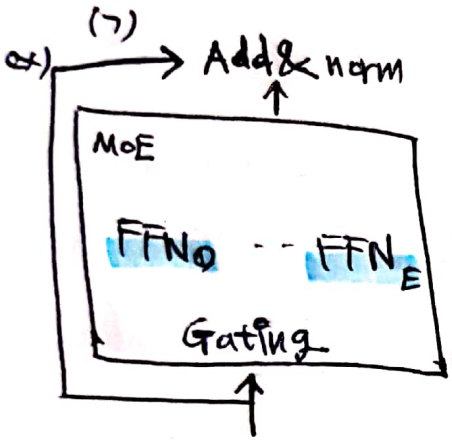
ECE 「expected calibration Error」

① Vanilla model 이란?

② perturbation

③ Top k- transformers.

⊕ 제플 리젠 여부X.?? <2>



MoE transformer Encoder

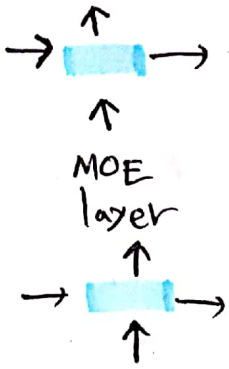
↳ 기존 transformer의

Multi-head Attention과

self

FF 사이에 (가)를 대입한다.

「Mixture-of-expert model = MoE layer」



↳each expert = usually resides at a separate GPU/server.

↳output

$=y=\sum_{i=1}^{n} G_i(x) \cdot E_i(x)$

$G_i(x)$: gating function, $E_i(x)$: expert output.

↳ Gating: $G(x) = \text{softmax}(\text{keep Top K}(H(x)))$

$[H(x)]_i = W_x + \text{noise}$

⇒ gating function are different from model to model.

「transformer models

= each FFN layer is replaced by MoE.」

Capacity Factor

↳ expert의 capacity 계산시 사용. Expert capacity allows more buffer to help mitigate token overflow during routing.

expert capacity = batch size of each expert.

↳ calculate : (tokens_per_batch / num_experts) x capacity_factor

pathways = A single model that can generalize across millions of tasks.