Special tokens in NLP

- Common special tokens in NLP

Token	Meaning / use
<pad>	Padding to make sequence the same lenght in a batch size
<eos>	End of sentence (marks where text stops) ⇒ stops GPT generation
<bos>	Beginning of sentence (used in some models to indicate start)
<unk>	Unknown token (used for words not in vocab)
<cls>	classification token (used in Bert for sentence lvl tasks) Ex i love Pizza ⇒ <cls> I love Pizza (bert would add cls to start) • cls represents whole sentence • after encoding, the vector for cls is used for tasks like, sentiment classification, topic classification etc.
<sep>	Seperator (splits sentences or segments in a sequence) helps seperate different segments in sentece like making QA pairs Ex <cls> how are you? <sep> i am fine <cls> → represents whole input for classification, <sep> → marks boundary between Q and A (sep = segment seperator)
<mask>	Masked tokens (used for masking tokens see p 125)

focal loss functions

[ex: Categorical Focal crossentropy loss]

Binary cross entropy ↘ Categorical crossentropy ↙

focal loss is a modification to standard cross entropy loss functions (mainly BCE / CCE) to specifically handle extream class impalance problems

- in a standard approach (CCE) the loss is for multiclass classification (ie lo img classes) CCE calculates error based on how far model predictions where from true lable it gives the same weight to all errors treating all easy and hard examples the same for ex. if 1 class has no defects in img the model quikly learns to guess no defect most of the time this massive number of correct predictions dominate total loss burying gradients needed to needed to learn defective classes
- the focal loss approach introduces a modulating factor $(1-p_t)^\gamma$ the cross entropy function this factor dynamically scales the contrabution of each example to the overall loss forcing model to prioratize learning from hard examples
  - p_t = predicted prob for correct class
  - γ (gamma usually 2.0) focus param controls down weighting

Ex:

EX type	model confidence p_t	modulating factor $(1-p_t)^\gamma$	Effect
easy (majority class)	High (ex 0.99)	very small (ex 0.01^2 ≈ 0.0001)	loss becomes negligible
hard (minority class)	low (ex 0.01)	close to 1 ex $0.9^2 \approx 0.81$	loss remains high forcing focus

Ex2: • true label ⇒ A

Model Pred Prob ⇒ A: 0.95, B: 0.03, C: 0.02

• p_t (prob true class) = 0.95

metric	calc	value
CCE loss	$-\log(p_t)$	$-\log(0.95) \approx 0.051$
Modulating factor	$(1-p_t)^\gamma$	$(1-0.95)^2 \approx 0.00013$
Focal loss	CCE loss × Factor	$0.051 \times 0.0025 \approx 0.00013$

Result: the model was super confident (high p_t) meaning it has already learned this example so we scale the loss down and vise versa so millons of easy ex dont dominate gradients (learning).