டிரான்ஸ்ஃபார்மரின் உள்ளமைப்பு: ஒரு விரிவான பார்வை

நாம் முன்பு, "I love cats" என்பதை "நான் பூனைகளை விரும்புகிறேன்" என்று மொழிபெயர்த்த கதையைப் பார்த்தோம். இப்போது, அந்த மாயாஜாலத்தின் பின்னணியில் உள்ள ஒவ்வொரு படியையும், அதன் கணிதத்தையும் ஆழமாக ஆராய்வோம்.

என்கோடரின் பயணம்: அர்த்தத்தை எண்களாக மாற்றுதல்

என்கோடரின் முக்கியப் பணி, உள்ளீட்டு வாக்கியத்தின் முழுமையான, சூழல் அறிந்த ஒரு பிரதிநிதித்துவத்தை உருவாக்குவது.

படி 1: உள்ளீட்டுப் பிரதிநிதித்துவம் (Input Representation)

முதலில், "I love cats" என்ற வாக்கியத்தின் ஒவ்வொரு வார்த்தைக்கும், அதன் அர்த்தத்தைக் குறிக்கும் எம்பெடிங் வெக்டரும் (e), அதன் இடத்தைக் குறிக்கும் இடக்குறியீடு வெக்டரும் (P) உருவாக்கப்பட்டு, இரண்டும் கூட்டப்படுகின்றன.

- "I" (Position 1) → $\mathbf{e}*1* + *mathbf{P}*1$
- "love" (Position 2) → $\mathbf{e}*2* + *mathbf{P}*2$
- "cats" (Position 3) → $\mathbf{e}*3* + *mathbf{P}*3$

இந்த இறுதி வெக்டர்கள்தான், என்கோடர் அடுக்கின் முதல் உள்ளீடு.

படி 2 & 3: கவனமும் சிந்தனையும் (Attention & FFN)

இந்த வெக்டர்கள், நாம் முன்பு விவாதித்த **Multi-Head Self-Attention** என்ற குழு உரையாடல் அறைக்கும், பின்னர் **Feedforward Neural Network (FFN)** என்ற தனிப்பட்ட சிந்தனைப் பட்டறைக்கும் அனுப்பப்படுகின்றன. இந்தச் சுழற்சி, பல அடுக்குகளில் மீண்டும் மீண்டும் நிகழும்போது, வார்த்தைகளின் பிரதிநிதித்துவம் மேலும் மேலும் செறிவூட்டப்படுகிறது.

இறுதியில், என்கோடர், உள்ளீட்டு வாக்கியத்தின் ஒவ்வொரு வார்த்தைக்கும், அதன் சூழலை முழுமையாக உணர்ந்த, ஒரு புதிய, ஞானம் பெற்ற வெக்டர்களின் தொகுப்பை வெளியீடாக அளிக்கிறது. இந்த அறிவுப் பெட்டகம்தான், டிகோடரின் பார்வைக்கு வைக்கப்படுகிறது.

டிகோடரின் பயணம்: புதிய மொழியில் படைத்தல்

டிகோடர், என்கோடரின் வெளியீட்டை வைத்துக்கொண்டு, வார்த்தைக்கு வார்த்தை, ஒரு புதிய வாக்கியத்தை உருவாக்குகிறது.

படி 1: முகமூடியணிந்த கவனம் (Masked Multi-Head Self-Attention)

டிகோடர், தனது முதல் வார்த்தையான "நான்" என்பதை உருவாக்கிய பிறகு, அடுத்த வார்த்தையை உருவாக்கும் முன், அது "நான்" என்பதை மட்டும் தனது கவனத்தில் கொள்ளும். எதிர்கால வார்த்தைகள் முகமூடியால் (Masking) மறைக்கப்படும்.

படி 2 & 3: என்கோடர்-டிகோடர் கவனமும், இறுதிச் செதுக்கலும்

அடுத்து, டிகோடர் தனது தற்போதைய நிலையில் இருந்து ஒரு கேள்வியை (Q) கேட்கிறது. அந்தக் கேள்வி, என்கோடர் வழங்கிய அறிவுப் பெட்டகத்தில் உள்ள ஒவ்வொரு வார்த்தையின் அடையாளம் (K) மற்றும் தகவலுடன் (V) உரையாடுகிறது. இந்த உரையாடலின் மூலம், மூல வாக்கியத்தின் எந்தப் பகுதிக்கு இப்போது கவனம் செலுத்த வேண்டும் என்பதைத் தீர்மானித்து, அதிலிருந்து பெற்ற தகவலை, **FFN** பட்டறைக்கு அனுப்பி, தனது அடுத்த வார்த்தையைத் தீர்மானிக்கிறது.

இந்தச் சுழற்சி, இறுதி வெளியீட்டை உருவாக்கும் வரை தொடரும்.

வெளியீட்டு அடுக்கு (Output Layer): எண்ணை வார்த்தையாக மாற்றுதல்

டிகோடர் அடுக்கின் இறுதி வெளியீடு (h), இன்னும் ஒரு கணித வெக்டர்தான். அதை நமக்குத் தெரிந்த ஒரு வார்த்தையாக மாற்றுவதற்கு, வெளியீட்டு அடுக்கு இரண்டு முக்கியப் பணிகளைச் செய்கிறது.

1. லீனியர் உருமாற்றம் (Linear Transformation):

டிகோடரின் இறுதி வெக்டர் (h), ஒரு எடை மேட்ரிக்ஸ் $(\mathbf{W}_o)$ மூலம் பெருக்கப்பட்டு, ஒரு புதிய வெக்டராக (z) மாற்றப்படுகிறது. இது, டிகோடரின் சிந்தனையை, அகராதியில் உள்ள எல்லா வார்த்தைகளுக்கான மதிப்பெண்களாக (scores) மாற்றுகிறது.

$$\mathbf{z} = \mathbf{W}_o\mathbf{h} + \mathbf{b}_o \tag{1}$$

2. சாஃப்ட்மேக்ஸ் செயல்பாடு (Softmax Function):

இந்த மதிப்பெண்கள், **Softmax** என்ற செயல்பாட்டின் மூலம், நிகழ்தகவுகளாக (Probabilities) மாற்றப்படுகின்றன. இது, ஒவ்வொரு வார்த்தைக்கும் 0 முதல் 1 வரை ஒரு நிகழ்தகவு மதிப்பைக் கொடுக்கும். மேலும், எல்லா வார்த்தைகளின் நிகழ்தகவுகளின் கூட்டுத்தொகை 1 ஆக இருக்கும்.

$$\text{Output} = \text{Softmax}(\mathbf{z}) \tag{2}$$

அதிக நிகழ்தகவு கொண்ட வார்த்தையே, டிகோடரின் அடுத்த வார்த்தையாகத் தேர்ந்தெடுக்கப்படுகிறது.

பயிற்சியின் ரகசியங்கள்: மாடலை மேம்படுத்தும் கருவிகள்

ஒரு டிரான்ஸ்ஃபார்மர், இவ்வளவு சிக்கலான பணிகளைச் சரியாகச் செய்ய, அதன் பயிற்சியின் போது இரண்டு முக்கியமான கருவிகள் உதவுகின்றன. அவை, மாதிரியின் கற்றல் திறனை நிலைப்படுத்துகின்றன.

1. லேயர் நார்மலைசேஷன் (Layer Normalization):

இது, ஒவ்வொரு அடுக்கின் வெளியீட்டையும் ஒரு குறிப்பிட்ட சீரான நிலைக்குக் கொண்டுவரும் ஒரு செயல். ஒரு நெடுஞ்சாலையில், வாகனங்கள் சீரான வேகத்தில் செல்வதை உறுதி செய்வது போல, இது தகவல்களின் ஓட்டத்தைச் சீரமைக்கிறது. இது, மாதிரியின் பயிற்சியை வேகமாகவும், நிலையானதாகவும் மாற்றுகிறது.

ஒரு உள்ளீடு x-இன் சராசரி (μ) மற்றும் மாறுபாடு (σ^2) ஆகியவற்றைக் கொண்டு, அது ஒரு புதிய வரையறைக்குள் கொண்டுவரப்படுகிறது. γ மற்றும் β என்பவை, இந்தச் சீரமைப்பை மேலும் நுணுக்கமாக மாற்ற, பயிற்சியின் போது கற்றுக்கொள்ளப்படும் அளவுருக்கள்.

$$\text{LayerNorm}(x) = \gamma \cdot \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{3}$$

2. ரெசிடுவல் இணைப்புகள் (Residual Connections):

இது, தகவல்கள் தொலைந்து போகாமல் இருக்க உதவும் ஒரு "தகவல் குறுக்குவழி" (information highway) ஆகும். ஒரு அடுக்கு, ஒரு உள்ளீட்டை (x) எடுத்து, அதைச் செயலாக்கி $(\text{Sublayer}(x))$ ஒரு வெளியீட்டை உருவாக்கும்போது, அந்த வெளியீட்டுடன், ursprüngliche உள்ளீட்டையும் (x) நேரடியாகக் கூட்டுகிறது.

$$\text{Output} = x + \text{Sublayer}(x) \tag{4}$$

இந்தக் குறுக்குவழி, பயிற்சியின் போது, சாய்வுகள் மறைந்துபோகும் (vanishing gradients) பிரச்சினையைத் தவிர்த்து, மிக ஆழமான அடுக்குகளுக்கும் கற்றலுக்கான சமிக்ஞை சரியாகச் செல்வதை உறுதி செய்கிறது.

கற்றல் செயல்முறை: தவறுகளிலிருந்து பாடம் பெறுதல்

டிரான்ஸ்ஃபார்மர், **குறுக்கு-என்ட்ரோபி இழப்பு (Cross-Entropy Loss)** என்ற முறையின் மூலம் பயிற்சி பெறுகிறது. இது, மாதிரி யூகித்த வெளியீட்டிற்கும் $(\hat{y}_i)$, உண்மையான சரியான வெளியீட்டிற்கும் (y_i) உள்ள வித்தியாசத்தைக் கணக்கிடுகிறது.

$$\text{Loss} = -\sum_{i=1}^{n} y_i \log(\hat{y}_i) \tag{5}$$

ஒவ்வொரு முறையும், இந்த இழப்பு மதிப்பைக் குறைக்கும் வகையில், மாதிரி தனது மில்லியன் கணக்கான அளவுருக்களைச் சரிசெய்துகொள்கிறது. இப்படி, பில்லியன் கணக்கான உதாரணங்களிலிருந்து, லட்சக்கணக்கான முறை தவறுகள் செய்து, அவற்றைத் திருத்திக்கொள்வதன் மூலமே, டிரான்ஸ்ஃபார்மர் மொழியின் ஆன்மாவைக் கற்றுக்கொள்கிறது.