1 Math and Probability

σ-algebra

$\mathcal{F} \subseteq \mathcal{P}(\Omega)$ is a σ-algebra over Ω iff 1. $\Omega \in \mathcal{F}$ 2. if $\varepsilon \in \mathcal{F}$, then $\varepsilon^{\mathsf{C}} \in \mathcal{F}$ 3. if $\varepsilon_1, \varepsilon_2 \ldots$ is a finite or infinite sequence in $\mathcal{F}$, then $\bigcup_n \varepsilon_n \in \mathcal{F}$

Measurable Space $(\Omega, \mathcal{F})$

Probability Measure

$\mathbb{P} : \mathcal{F} \to [0,1]$ where 1. $\mathbb{P}(\Omega) = 1$ 2. If $\varepsilon_1, \varepsilon_2$ is a countable sequence of disjoint sets in $\mathcal{F}$, then $\mathbb{P}(\bigcup_n \varepsilon_n) = \sum_n \mathbb{P}(\varepsilon_n)$

Probability Space $(\Omega, \mathcal{F}, \mathbb{P})$

Algebra

$\mathcal{A} \subseteq \mathcal{P}(\Omega)$ over Ω if 1. $\Omega \in \mathcal{A}$ 2. $\varepsilon \in \mathcal{A} \implies \varepsilon^{\mathsf{C}} \in \mathcal{A}$ 3. $\varepsilon_1, \varepsilon_2 \in \mathcal{A} \implies \varepsilon_1 \cup \varepsilon_2 \in \mathcal{A}$

Probability Pre-Measure

$\mathbb{P}_0 : \mathcal{A} \to [0,1]$ s.t. 1. $\mathbb{P}_0(\Omega) = 1$ 2. If $\varepsilon_1, \varepsilon_2, \ldots$ is a countable sequence of disjoint sets in $\mathcal{A}$ *whose countable union is also in* $\mathcal{A}$, then $\mathbb{P}_0(\bigcup_{n=1}^{\infty} \varepsilon_n) = \sum_{n=1}^{\infty} \mathbb{P}_0(\varepsilon_n)$

Geometric Series

$S = \sum_{n=0}^{\infty} ar^n = \frac{a}{1-r}, \quad \text{for } |r| < 1$

Cylin $\overline{C}(\mathcal{H}) = \{\boldsymbol{y}\boldsymbol{\omega} | \boldsymbol{y} \in \mathcal{H}, \boldsymbol{\omega} \in \overline{\Sigma}^{\infty}\}$

Binomial

$\mu = \mathbb{E}[x] = np$, $\sigma^2 = np(1-p)$

Gaussian 68, 95, 99.7

$p(x) = \frac{1}{\sqrt{2\sigma^2}} \exp\{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2\}$

Triangle Inequality

$\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$

Matrix Multiplication

$\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{B} \in \mathbb{R}^{n \times p}$ $\mathbf{AB} \in \mathbb{R}^{m \times p}$ time $\mathcal{O}(m \times n \times p)$

Cosine similarity $\frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$

softmax$(\mathbf{x})_d = \frac{\exp\{x_d\}}{\sum_{j=1}^{D} \exp\{x_j\}}$

2 Foundations

Sequence Model

A probability space over the set $\Sigma^* \cup \Sigma^{\infty}$

Language Model

A discrete distribution p_{LM} over Σ^* or $\mathbb{P} * (\Sigma^{\infty}) = 0$

GNM $p_{\mathbf{LM}}(\boldsymbol{y}) = \frac{\exp\{-\hat{p}_{\mathbf{GN}}(\boldsymbol{y})\}}{\sum_{\boldsymbol{y}' \in \Sigma^*} \exp\{-\hat{p}_{\mathbf{GN}}(\boldsymbol{y}')\}}$

LNM

$p_{\text{LN}}(\boldsymbol{y}) = p_{\text{SM}}(\textsc{eos}|\boldsymbol{y}) \prod_{t=1}^{T} p_{\text{SM}}(y_t|\boldsymbol{y}_{<t})$

Prefix Prob $\pi(\boldsymbol{y}) = \sum_{\boldsymbol{y}' \in \Sigma^*} p_{\mathbf{LM}}(\boldsymbol{y}\boldsymbol{y}')$

Probability Measure of LNM

1. Define $\overline{\mathcal{C}} \subseteq \mathcal{P}(\Omega)$ as an algebra over $\Omega = \overline{\Sigma}^{\infty}$ where $\overline{\mathcal{C}} = \bigcup_{k=1}^{\infty} \overline{\mathcal{C}}_k$

2. $\mathbb{P}_0(\overline{C}(\mathcal{H})) = \sum_{\overline{\boldsymbol{y}} \in \mathcal{H}} p_{LN}(\overline{\boldsymbol{y}})$

3. Extend $\mathbb{P}_0$ to $(\overline{\Sigma}^{\infty}, \sigma(\overline{\mathcal{C}}), \mathbb{P})$

4. Construct SM: $\mathcal{C}(\mathcal{H}) = \{\boldsymbol{y}\boldsymbol{\omega} | \boldsymbol{y} \in \mathcal{H}, \boldsymbol{\omega} \in \Sigma^* \cup \Sigma^{\infty}\}$ then, $x(\boldsymbol{\omega}) = \begin{cases} \boldsymbol{\omega}_{<k} & \text{if } k \text{ is the first } \textsc{eos} \text{ in } \boldsymbol{\omega} \\ \boldsymbol{\omega} & \text{otherwise} \end{cases}$

Tightness

An LNM is tight **IFF** $\tilde{p}_{\text{EOS}}(t) = 1$ for some t or $\sum_{t=1}^{\infty} \tilde{p}_{\text{EOS}}(t) = \infty$

3 Classical LMs

FSA $\mathcal{A} = (Q, \Sigma, \delta, I, F)$

WFSA $\mathcal{A} = (Q, \Sigma, \delta, \lambda, \rho)$

Deterministic FSA

1. No ε-transitions 2. $\forall (q,a) \in Q \times \Sigma$, at most one $q' \in Q$ s.t. $q \xrightarrow{a} q' \in \delta$ 3. $|I| = 1$.

$w(\pi) = \lambda(q_1) \prod_{i=1}^{N} w_i \rho(q_N)$.

$\mathcal{A}(\boldsymbol{y}) = \sum_{\pi \in \Pi(A,\boldsymbol{y})} w(\pi)$

$Z(\mathcal{A}) = \sum_{\boldsymbol{y} \in \Sigma^*} \mathcal{A}(\boldsymbol{y})$

Probabilistic FSA

λ, ρ and the weights are non-negative, $\sum_{q \in Q} \lambda(q) = 1$, $\forall q \in Q$ we have $\rho(q) + \sum_{q \xrightarrow{a/w} q'} w = 1$

Tightness of PFSA

A PWFSA is **tight** if and only if all accessible states are co-accessible.

Finite State LM

$\exists \mathcal{A} = (\Sigma, Q, \delta, \lambda, \rho)$, $L(\mathcal{A}) = L(p_{\text{LM}})$

n-gram assumption

$p_{\text{SM}}(y_t|\boldsymbol{y}_{<t}) = p_{\text{SM}}(y_t|y_{t-1} \cdots y_{t-n+1})$ PAD with BOS $n-1-t$ times. $\mathcal{O}(|\Sigma|^{n-1})$

Repre. based n-gram

$\boldsymbol{\theta} = \{\theta_{y|\boldsymbol{y}} = p_{\text{SM}}(y|\boldsymbol{y}) \mid y \in \overline{\Sigma}, \boldsymbol{y} \in \overline{\Sigma}^{n-1}, \theta_{y|\boldsymbol{y}} \geq 0, \sum_{y' \in \overline{\Sigma}} \theta_{y'|\boldsymbol{y}} = 1\}$

MLE $p_{\mathbf{SM}}(y_n|\boldsymbol{y}_{<n}) = \frac{C(y_1, \ldots, y_n)}{C(y_1, \ldots, y_{n-1})}$

Bengio's Model $p_{\mathbf{SM}}(\overline{y}_t|\overline{\boldsymbol{y}}_{<t}) = \text{softmax}(\textsc{enc}(\boldsymbol{y}_{t-1:t-n+1})^\top \mathbf{E} + \mathbf{b})_{\overline{y}_t}$

CFG $\mathcal{G} = (\Sigma, \mathcal{N}, S, \mathcal{P})$

WCFG $\mathcal{W} : \mathcal{P} \to \mathbb{R}$

PCFG

$\mathcal{W}$ is non-negative, $\forall X \in \mathcal{N}$ we have $\sum_{X \to \alpha \in \mathcal{P}} \mathcal{W}(X \to \alpha) = 1$.

WCFG Allsum

$Z(\mathcal{G}) = \sum_{d \in \mathcal{D}_{\mathcal{G}}} w(d)$

$= \sum_{d \in \mathcal{D}_{\mathcal{G}}} \prod_{(X \to \alpha) \in d} \mathcal{W}(X \to \alpha)$

Tightness of PCFG

For a PCFG $\mathcal{G}$ with $|\mathcal{N}| = N$ we define for each $X_n \in \mathcal{N}$ its production generating fct. $g_n\left((s_i)_{i=1}^N\right) = \sum_{X_n \to \alpha} \mathcal{W}(X_n \to \alpha) s_1^{r_1(\alpha)} \cdots s_N^{r_N(\alpha)}$ where $r_i(\alpha)$ is the number of times X_i appears in α. Then we set $E \in \mathbb{R}^{N \times N}$ to have entries $E_{nm} = \frac{\partial}{\partial s_m} g_n(s_1, \ldots, s_N)\big|_{s_1, \ldots, s_N = 1}$. Then $\mathcal{G}$ is **tight** if $\lambda < 1$ and non-tight if $\lambda > 1$, where $\lambda = \max\left(\{|\lambda'| \mid \lambda' \in \sigma(E)\}\right)$.

Pushdown Automaton

A language is context-free IFF it is recognized by some PDA.

Multi-Stack PDA

Any (probabilistic) 2-stack PDA is Turing complete. Hence, the tightness of a probabilistic 2-stack PDA is undecidable.

4 RNNs

RNN

A RNN is given by an initial state $h_0 \in \mathbb{R}^d$ and a dynamics map $h_t = f(h_{t-1}, y_t)$. An RNN-LM uses $\text{enc}(y_{<t+1}) = h_t$, $E \in \mathbb{R}^{|\overline{\Sigma}| \times d}$

Recurrent Neural Sequence Model

$p_{\mathbf{SM}}(y_t|\boldsymbol{y}_{<t}) = \mathbf{f}_{\Delta^{|\overline{\Sigma}|-1}}(\mathbf{Enc}_{\mathcal{R}}(\boldsymbol{y}_{<t}))_{y_t}$

Elman RNN

An Elman RNN is an RNN with $f(h_{t-1}, y_t) = \sigma(U h_{t-1} + V e'(y_t) + b)$, where $U \in \mathbb{R}^{d \times d}$, $V \in \mathbb{R}^{d \times R}$ and $b \in \mathbb{R}^d$ and $e' : \Sigma \to \mathbb{R}^R$ is an input embedding function.

Jordan RNN $f(h_{t-1}, y_t) = \sigma(U \sigma'(E h_{t-1}) + V e'(y_{t-1}) + b)$

Tightness of RNN-LMs

If the LM uses the softmax and $s\|h_t\| \leq \log t$ (in particular if f is bounded, e.g. if f uses a bounded activation function), then the induced LM is **tight**.

Expressiveness of RNNs

HRRNs (over $\overline{\mathbb{R}}$) $\equiv$ dPFSA for any activation function with finite image. Minsky's construction encodes any dPFSA using $U \in \mathbb{R}^{|\Sigma||Q| \times |\Sigma||Q|}$, $U_{n(q',y'),n(q,y)} = \mathbb{I}\{q_t \xrightarrow{y'/\circ} q' \in \delta\}$ to encode which states are reachable from h_{t-1} $V \in \mathbb{R}^{|\Sigma||Q| \times |\Sigma|}$, $V_{n(q',y'),m(y')} = \mathbb{I}\{\circ \xrightarrow{y'/\circ} q' \in \delta\}$ to encode which states can be transitioned to using y_t.

$$E_{\overline{m}(y')n(q,y)} = \begin{cases} \log \omega(q \xrightarrow{y'/w} \circ) & \text{if } y' \in \Sigma \\ \log \rho(q) & \text{otherwise} \end{cases}$$

Can reduce the hidden state dimensionality to $\Omega(|\Sigma|\sqrt{|Q|})$.

Saturated Sigmoid Elmann RNNs are Turing complete (because they can encode two-stack PDAs). It is thus undecidable whether an RNN-LM is **tight**.

5 Transformers

Attention

$f : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$, $\mathbf{q} \in \mathbb{R}^D$, $\mathbf{K}_t \in \mathbb{R}^{t \times D}$, $\mathbf{V}_t \in \mathbb{R}^{t \times D}$. $\mathtt{Att}(\mathbf{q}_t, \mathbf{K}_t, \mathbf{V}_t) : \mathbb{R}^D \times \mathbb{R}^{t \times D} \times \mathbb{R}^{t \times D} \to \mathbb{R}^D$

$\mathbf{s}_t = \mathbf{f}_{\Delta^{D-1}}(f(\mathbf{q}, \mathbf{k}_1), \ldots, f(\mathbf{q}, \mathbf{k}_t))$

$\mathbf{a}_t = \mathtt{Att}(\mathbf{q}_t, \mathbf{K}_t, \mathbf{V}_t) = s_1 \mathbf{v}_1 + \cdots + s_t \mathbf{v}_t$

Transformer Layer

$\mathtt{T} : \mathbb{R}^{T \times D} \to \mathbb{R}^{T \times D}, \mathbf{X}, \mathbf{Z} \in \mathbb{R}^{T \times D}$

$\mathbf{a}_t = \mathtt{Att}(Q(\mathbf{x}_t), K(\mathbf{X}_t), V(\mathbf{X}_t)) + \mathbf{x}_t$

$\mathbf{z}_t = O(\mathbf{a}_t) + \mathbf{a}_t$

$\mathtt{T}(\mathbf{X}) = \mathbf{Z} = (\mathbf{z}_1^\top, \mathbf{z}_2^\top, \ldots, \mathbf{z}_T^\top) \in \mathbb{R}^{T \times D}$

Transformer

$\mathbf{X}^1 = (\mathbf{e}'(y_0), \mathbf{e}'(y_1), \ldots, \mathbf{e}'(y_t))$

$\mathbf{Z}^\ell = \mathtt{T}_\ell(\mathbf{X}^\ell) \quad \text{for } 1 \leq \ell < L$

$\mathbf{X}^{\ell+1} = \mathbf{Z}^\ell, \mathbf{h}_t = F(\mathbf{z}_t^L)$

Attention Block $\mathtt{A} : \mathbb{R}^{T \times D} \to \mathbb{R}^{T \times D}$

$\mathtt{A}(\mathbf{X}) = \mathbf{f}_{\Delta^{D-1}}\left(Q(\mathbf{X}) K(\mathbf{X})^\top\right) V(\mathbf{X})$

$\mathbf{U} = Q(\mathbf{X}) K(\mathbf{X})^\top \in \mathbb{R}^{T \times T}$.

$W_i^Q, W_i^K \in \mathbb{R}^{d \times d_k}$, $W_i^V \in \mathbb{R}^{d \times d_v}$

$W^O \in \mathbb{R}^{h d_v \times d}$ often $d_k = d_v = \frac{d}{h}$

Masked Attention Block

$\mathtt{A}(\mathbf{X}, \mathbf{M}) = \text{softmax}(Q(\mathbf{X}) K(\mathbf{X})^\top \odot \mathbf{M}) V(\mathbf{X})$

$M_{i,j} = \mathbb{I}[i \leq j] + -\infty \mathbb{I}[i > j]$

Positional Enc $\mathbf{f}_{\mathbf{pos}} : \mathbb{N} \to \mathbb{R}^D$

$\mathbf{e}'_{\text{pos}}(y_t) = \mathbf{e}'(y_t) + \mathbf{f}_{\text{pos}}(t)$

$\mathbf{e}'_{\text{pos}} : \overline{\Sigma} \to \mathbb{R}^D$

MH-A

$\mathbf{f}_H : \mathbb{R}^{T \cdot H \times D} \to \mathbb{R}^{T \times D}$ $\mathtt{MH\text{-}A}(\mathbf{X}) = \mathbf{f}_H(\text{concat}_{0 \leq h < H}(\text{softmax}(Q_h(\mathbf{X}) K_h(\mathbf{X})^\top) V_h(\mathbf{X})))$

Layer Norm $\mathtt{LN} : \mathbb{R}^D \to \mathbb{R}^D$

$\mathtt{LN}(\mathbf{x}; \boldsymbol{\gamma}, \boldsymbol{\beta}) = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\sqrt{\sigma^2(\mathbf{x}) + \epsilon}} \odot \boldsymbol{\gamma} + \boldsymbol{\beta}$

FFN

$\text{FFN}(x) = \max(0, x W_1 + b_1) W_2 + b_2$

Tightness of Transformers

Any transformer using soft attention is **tight** (because its layers are continuous and the set of possible inputs to the first layer is compact, making enc bounded).

Expressiveness of Transformers

Let p_{LN} be an n-gram language model. Then, there exists a transformer $\mathcal{T}$ with $L(p_{LN}) = L(\mathcal{T})$.

6 Tokenization

Tokenizer

A tokenizer model from Σ^* to Δ^* is a pair of stochastic maps $T = (\tau, \kappa)$, $\tau : \Sigma^* \mapsto \Delta^*, \kappa : \Delta^* \mapsto \Sigma^*$

BPE

Input: Σ, $C = \{x^{(m)}\}_{m=1}^M \subset \Sigma^*$. Return: $\Delta, t : \Sigma^* \to \Delta^*$, Initialize Δ to Σ, Find the most frequent merge in C, where a merge m is a concatenation of two elements in Δ, so now $\Delta \leftarrow \Delta \cup m$.

Suprious Ambiguity

aaba $\to$ |a|a|b|a| or |aa|b|a|

Pushforward

$p(x) = \sum_{y \in t^{-1}(x)} p_{\Delta^k}(y)$

$t^{-1}(x) = \{y | y \in \Delta^*, t(x) = y\}$

7 Sampling

Ancestral Sampling

1. Locally normalize.
2. Sample $y_t \sim p(\cdot \mid y_{<t})$, stop when $y_t = \text{EOS}$.

May not halt $\rightarrow$ set max string length.

Greedy

$x_i = \text{argmax}_{x \in \Sigma^*} \log(x | x_1 \dots x_{i-1})$

Sampling Adaptors

To calibrate p we can postprocess probabilities by a function $\alpha : \Delta^{|\Sigma|-1} \rightarrow \Delta^{|\Sigma|-1}$.

Top-K Sampling

Set $p(y_t \mid y_{<t}) = 0$ for all but the K most probable tokens, and renormalize.

Nuclues Sampling

Only take top $p\%$ of probability mass.

8 Transfer Learning

ELMo

Fwd & Bwd LM using L LSTM layers. The ELMo representation for a token y_t is $\textbf{ELMo}^{\text{task}} = \gamma^{\text{task}} \sum_{l=0}^{L} s_l^{\text{task}} \mathbf{h}_{tl}^{\text{LM}}$ where $s_l^{\text{task}} \geq 0$, $\mathbf{h}_{tl}^{\text{LM}} = (\overrightarrow{h}_{tl}^{LM}, \overleftarrow{h}_{tl}^{LM})$.

BERT (encoder) $\mathcal{L} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{NSP}}$

$\mathcal{L}_{\text{MLM}}(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T}$

$\log_{\text{MLM}}(y_t^{(i)} | \boldsymbol{y}_{<t}^{(i)}, \boldsymbol{y}_{>t}^{(i)}; \theta) \mathbb{I}\{y_t^{(i)} = \text{M}\}$

9 Parameter Efficient Fine-Tuning

BitFit

$\mathbf{Q}(\mathbf{x}) = \mathbf{W}_q \mathbf{x} + \mathbf{b}_q$

$\mathbf{h}_4 = \text{GELU}(\mathbf{W}_2 \cdot \mathbf{h}_3 + \mathbf{b}_2)$

Diff Pruning

$\theta_{\text{FT}} = \theta_{\text{LM}} + \delta$. Encourage δ_{Diff} to be sparse by regularization by a differentiable approximation to the L_0-norm penalty as $\|\delta_{Diff}\|_0$.

Adapters

Insert bottleneck MLPs after each sublayer (MHA and FFN).

$\mathbf{h} \leftarrow \mathbf{h} + f(\mathbf{h}\mathbf{W}_{\text{down}})\mathbf{W}_{\text{up}}$

LoRA

Replace weight matrices $W \in \mathbb{R}^{d \times r}$ with $W \leftarrow W + \frac{\beta}{b} AB$ where $A \in \mathbb{R}^{d \times b}$ and $B \in \mathbb{R}^{b \times r}$ are random matrices and β is a constant in b.

$N_{\text{param}} = NH(3b(d+r) + 2bd)$

10 Prompting

Objective

$\hat{z} = \underset{z \in \mathcal{Z}}{\text{search}} P(f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z}); \theta)$.

Discrete Prompts

Use the middle words/paths as templates in the form of `[X] middle words [Z]`, translating the prompt into another language and back, use a thesarus to replace words, training a text generation model for generating prompts

Prefix Tuning

Prepends a sequence of continuous task-specific vectors to the input while keeping the LM parameters frozen. $\max_\phi \log P(\boldsymbol{y} | \boldsymbol{x}; \theta; \phi) = \max_\phi \sum_{y_i} \log P(y_i | h_{<i}; \theta; \phi)$

Self-Consistency

Generate multiple reasoning paths and selecting the most frequent final answer.

11 Vision Language Models

Vision Encoders

Pretrained **OD** then use visual features as well as location features $[x_1, y_1, x_2, y_2, w, h, w \times h]$; **CNN**; **Image Transformers** Create image tokens: Split image into image patches, map them into vectors and linearly project them to patch embeddings. Add a learnable special token [CLS]

Multimodal Fusion

Fusion encoder takes both $v = \{v_1, \cdots, v_M\}$ and $w = \{w_1, \cdots, w_N\}$ as input, and learns contextualized multimodal representations $\tilde{v} = \{\tilde{v}_1, \cdots, \tilde{v}_M\}$ and $\tilde{w} = \{\tilde{w}_1, \cdots, \tilde{w}_N\}$. **merged attention** (concat v, w), **co-attention** v, w are fed into different Transformer blocks independently.

Masked Language Modelling

$\mathcal{L}_{\text{MLM}}(\theta) = -\mathbb{E}_{(\tilde{\mathbf{w}}, \tilde{\mathbf{v}}) \sim D} \log P_\theta(\tilde{\mathbf{w}}_{\mathbf{m}} | \tilde{\mathbf{w}}_{\setminus \mathbf{m}}, \tilde{\mathbf{v}})$

Image Text Matching

$\mathcal{L}_{\text{ITM}}(\theta) = -\mathbb{E}_{(\tilde{\mathbf{w}}, \tilde{\mathbf{v}}) \sim D} [y \log s_\theta(\tilde{\mathbf{w}}, \tilde{\mathbf{v}}) + (1-y) \log(1 - s_\theta(\tilde{\mathbf{w}}, \tilde{\mathbf{v}}))]$

Image-Text Contrastive Learning

$s_{i,j}^{i2t} = v_i^\top w_j$, $s_{i,j}^{t2i} = w_i^\top v_j$

$\mathcal{L}_{\text{ITC}}^{i2t}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(s_{i,i}^{i2t}/\sigma)}{\sum_{j=1}^{N} \exp(s_{i,j}^{i2t}/\sigma)}$

$\mathcal{L}_{\text{ITC}}^{t2i}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(s_{i,i}^{t2i}/\sigma)}{\sum_{j=1}^{N} \exp(s_{i,j}^{t2i}/\sigma)}$

Masked Image Modelling

$\mathcal{L}_{\text{MIM}}(\theta) = \mathbb{E}_{(\tilde{\mathbf{w}}, \tilde{\mathbf{v}}) \sim D} P_\theta(\tilde{\mathbf{v}}_{\mathbf{m}} | \tilde{\mathbf{v}}_{\setminus \mathbf{m}}, \tilde{\mathbf{w}})$

12 RAG

TF-IDF

$\text{tf-idf}(t, d, \mathcal{D}) = \text{tf}(t, d) \times \text{idf}(t, \mathcal{D})$

$\text{tf}(t, d) = \log(1 + \text{freq}(t, d))$

$\text{idf}(t, \mathcal{D}) = \log\left(\frac{|\mathcal{D}|}{|d \in \mathcal{D} : t \in d|}\right)$

Scoring

$\text{score}(\mathbf{q}, \mathbf{d}) = \sum_{t \in q} \frac{\text{tf}(t,d)}{|d|}$

Dense Retrieval

$\text{SIM}(q, d) = \text{ENC}(q)^T \cdot \text{DEC}(d)$.

$L(q_i, d_i^+, d_{i,1}^-, \dots, d_{i,n}^-) = -\log \frac{e^{\text{SIM}(q_i, d_i^+)}}{e^{\text{SIM}(q_i, d_i^+)} + \sum_{j=1}^{n} e^{\text{SIM}(q_i, d_{i,j}^-)}}$

kNN-LM

Store all embedded prefixes and their following words in a database. At inference time, retrieve the kNN of a prefix, normalize exponentiated distances to a pron distr p_ξ over words. Then sample from a convex combination of p_ξ and the LM.

Dynamic Gating: Set the weighting of distributions depending on the prefix.

13 RLHF

1. Collect a dataset of instructions+answers and train a supervised baseline model.
2. Produce a dataset of comparisons of different answers given by the baseline model, score them manually and train a reward model.
3. Use PPO to fine-tune a LM (the policy) using the reward model.

14 Callibration

$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{M} |\text{acc}(B_m) - \text{conf}(B_m)|$

$\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{I}(\hat{y}_i = y_i)$, $\text{conf}(B_m) = \sum_{i \in B_m} \hat{p}_i$

15 Security & Adversarial examples

Greedy Coordinated ∇-Descent

1. Find top-k token substitutions according to gradient 2. Pick B substitutions at random across all suffix tokens 3. Evaluate the loss of all B candidates and pick the best.

Watermarking

Bias LLM away from true $p(s_{i+1} | s_i)$ in a subtle but verifiable way.

16 Prompt Injections

Indirect Attacks

1. The adversary plants *indirect prompts* on a source. 2. LLM retrieves the poisoned prompt.

Defenses

Escape data; Detect with 2nd LLM; Separate pipelines; instruction hierarchy; **quarantined LMs**.

17 Data poisoning, backdoors and model stealing

Poisoning Wikipedia

1. Estimate when each article was snapshot in the past dump. 2. Poison each article right before it.

Defenses

Integrity Check (but many FP), randomize snapshot times and keep edits longer than some time only.

Distillation

Train your own LLM to replicate the behavior of the original LLM.

Cryptanalysis

Dream of viewing LLM operations as cryptographic mechanisms and precisely recovering everything.

Recovering hidden dim

$\text{LLM}(\mathbf{x}_i) = \mathbf{y}_i = \mathbf{z}_i \mathbf{W}^\top$, so $\mathbf{Y} = \mathbf{Z}\mathbf{W}^\top$, $\mathbf{Y} \in \mathbb{R}^{n \times V}$, $\mathbf{Z} \in \mathbb{R}^{n \times h}$, $\mathbf{W}^\top \in \mathbb{R}^{h \times V}$, h is rank of $\mathbf{Y}$. Can recover part of weights using SVD $\mathbf{Y} = \mathbf{U}\Sigma\mathbf{V}^\top$, $\mathbf{U} \in \mathbb{R}^{n \times h}, \Sigma \in \mathbb{R}^{h \times h}$, $\mathbf{V}^\top \in \mathbb{R}^{h \times V}$, where $\mathbf{V}^\top$ are the weights up to an $h \times h$ transform. In practice, you only get top-k tokens, you can attack OpenAI's via `logit_bias` by repeatedly setting the top-k to $-\infty$ and push other values upwards.

18 Privacy

Federated Learning

Central server aggregates gradient updates from multiple clients. Issue: gradients are not private! Given a gradient g find $x_1 \dots x_B$ to minimize $\|g - \frac{1}{B} \sum_{i=1}^{B} \nabla_\theta \mathcal{L}(f_\theta(x_i))\|$

Weight-Trap Attack

Server sends client model f_θ s.t. $\nabla_\theta \mathcal{L}(f_\theta(x_i)) = x_i$

Differential Privacy

An algorithm M is ε-differentially private if for any "neighboring" databases D_1, D_2 that differ in a single element, and any output S we have: $P[M(D_1) \in S] \leq \exp(\varepsilon) P[M(D_2) \in S]$ Post-Processing: If M is ε-DP, then $f(M)$ for any function f is also ε-DP.

Composition: If M_1 is ε_1-DP and M_2 is ε_2-DP then $f(M_1, M_2)$ is $(\varepsilon_1 + \varepsilon_2)$-DP.

Sensitivity

$\Delta f = \max |f(\mathcal{D}_1) - f(\mathcal{D}_2)|$, release $y \sim \text{Laplace}(f(\mathcal{D}), \frac{\Delta f}{\varepsilon})$

Noisy Gradient Descent

$\mathbf{g} = \frac{1}{k} \sum_{x_i \in B} \nabla_\theta \mathcal{L}(f_\theta; \mathbf{x}_i) + \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ giving us a relaxed notion of DP with prob $1 - \delta$. Bound sensitivity by, clipping gradients to norm C. The entire training algorithm is then ε'-DP for some $\varepsilon' > \varepsilon$. Subsampling amplifies privacy: the gradient is $\approx (k\varepsilon)/(|D|)$-DP w.r.t. neigboring datasets.The final DP budget is in $(O)(\sqrt{N}\varepsilon)$.