# Aprendizado Profundo

**Formulário, Unidades 4 e 5**

Prof. Lucas S. Kupssinskü

Transformers e Modelos Geradores

---

*Este formulário reúne as principais expressões das unidades 4 e 5 e resultados matemáticos de apoio. Notação: $\mathcal{N}(\mu, \Sigma)$ denota a gaussiana de média $\mu$ e covariância $\Sigma$; $\odot$ é o produto elemento a elemento; $\sigma(\cdot)$ é a função sigmoide.*

## Matemática Básica

**Sigmoide e derivada:**

$$\sigma(a) = \frac{1}{1 + e^{-a}}, \qquad \sigma'(a) = \sigma(a)\big(1 - \sigma(a)\big).$$

**Softmax** (sobre *logits* $\mathbf{z} \in \mathbb{R}^K$, com temperatura $\tau$; $\tau = 1$ recupera a forma padrão):

$$\text{softmax}(\mathbf{z}/\tau)_i = \frac{e^{z_i/\tau}}{\sum_{j=1}^{K} e^{z_j/\tau}}.$$

**Logaritmos:**

$$\ln(ab) = \ln a + \ln b, \qquad \ln \frac{a}{b} = \ln a - \ln b, \qquad \ln a^b = b \ln a.$$

**Divergência de Kullback-Leibler** (caso contínuo):

$$D_{KL}[P \,\|\, Q] = \int P(x)\, \ln \frac{P(x)}{Q(x)}\, dx.$$

**Traço** (soma da diagonal principal) **e determinante de matriz diagonal:**

$$\text{Tr}[\text{diag}(d_1, \ldots, d_n)] = \sum_{i=1}^{n} d_i, \qquad \det[\text{diag}(d_1, \ldots, d_n)] = \prod_{i=1}^{n} d_i, \qquad \log\det = \sum_{i=1}^{n} \log d_i.$$

**Determinante de matriz $2 \times 2$ geral:**

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc.$$

## Atenção e Transformers

**Projeções** (entrada $X$ com um *token* por linha; $d$ é a dimensão das projeções):

$$Q = XW_q, \qquad K = XW_k, \qquad V = XW_v.$$

**Scaled dot-product attention:**

$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V.$$

**Máscara causal** (aplicada aos *scores* antes do *softmax*; posições com $-\infty$ recebem peso 0):

$$\text{score}_{ij} \leftarrow -\infty \quad \text{se } j > i.$$

**Multi-head attention** ($H$ cabeças em paralelo, cada uma com dimensão $d_{\text{model}}/H$; concatenação seguida da projeção de saída $W_O$):

$$\text{MultiHead}(Q, K, V) = \text{Concat}\big(\text{head}_1, \ldots, \text{head}_H\big)\, W_O, \qquad \text{head}_i = \text{attention}\big(QW_Q^i,\, KW_K^i,\, VW_V^i\big).$$

**Complexidade da *self-attention*** no comprimento de sequência $n$: $O(n^2 d)$.

**Layer normalization e conexão residual** (média e desvio padrão sobre a dimensão das *features*; $\gamma$ e $\beta$ treináveis):

$$\text{LayerNorm}(x) = \gamma \, \frac{x - \mu}{\sigma + \epsilon} + \beta, \qquad \text{output} = \text{LayerNorm}\big(x + \text{Subblock}(x)\big).$$

**Positional encoding sinusoidal** (posição $i$, dimensões $2k$ e $2k{+}1$; somado ao *embedding*):

$$\text{PE}(i, 2k) = \sin\!\left(\frac{i}{10000^{2k/d}}\right), \qquad \text{PE}(i, 2k{+}1) = \cos\!\left(\frac{i}{10000^{2k/d}}\right).$$

## Tokenização

---

**Score de merge do WordPiece** (reconstrução popularizada pelo Hugging Face; o BPE, em contraste, mescla o par adjacente de maior $\text{freq}(xy)$):

$$\text{score}(x, y) = \frac{\text{freq}(xy)}{\text{freq}(x) \cdot \text{freq}(y)}.$$

**Modelo de linguagem unigrama** (segmentação $x = (x_1, \ldots, x_n)$; probabilidades iniciais por contagem de substrings, com $N = \sum_{t'} \text{count}(t')$):

$$P(x) = \prod_{i=1}^{n} P(x_i), \qquad P(t) = \frac{\text{count}(t)}{N}.$$

**Viterbi** (melhor segmentação por programação dinâmica; $\delta(j)$ é o log da melhor segmentação do prefixo com $j$ caracteres, $s_{i:j}$ é o trecho entre as posições $i$ e $j$, $V$ é o vocabulário):

$$\delta(0) = 0, \qquad \delta(j) = \max_{\substack{i<j \\ s_{i:j} \in V}} \big[\, \delta(i) + \ln P(s_{i:j}) \,\big].$$

**Log-likelihood do corpus e critério de poda do Unigram** (a perda $\Delta_t$ mede quanto $\mathcal{L}$ cai ao remover o token $t$ e re-segmentar; podam-se os tokens de menor $\Delta_t$):

$$\mathcal{L} = \sum_{w \in \text{corpus}} \text{freq}(w) \, \ln P(\text{melhor seg. de } w), \qquad \Delta_t = \mathcal{L} - \mathcal{L}_{\text{sem } t}.$$

**Tabela de embedding** (lookup da linha $i$ da matriz $E$; $|V|$ é o tamanho do vocabulário):

$$\mathbf{e}_i = E[i], \quad E \in \mathbb{R}^{|V| \times d}, \qquad \#\text{params}_{\text{embedding}} = |V| \times d.$$

## Treinamento e Inferência de LLMs

---

**Loss de modelagem de linguagem** (entropia cruzada autorregressiva sobre $m$ *tokens*):

$$\mathcal{L} = -\sum_{t=1}^{m} \log P\big(y_t \mid y_{<t}\big).$$

**Lei de escala de Chinchilla** (volume ótimo de dados $D^*$, em *tokens*, para um modelo de $N$ parâmetros):

$$D^* \approx 20\, N.$$

**LoRA** (matriz congelada $W_0 \in \mathbb{R}^{d \times k}$; apenas $B \in \mathbb{R}^{d \times r}$ e $A \in \mathbb{R}^{r \times k}$ são treináveis):

$$h = W_0 \, x + \frac{\alpha}{r} \, BA \, x, \qquad \#\text{params treináveis} = r \, (d + k).$$

**Decodificação especulativa** (rascunho $q$, alvo $p$; *token* proposto $\tilde{x}$):

$$P(\text{aceitar } \tilde{x}) = \min\left(1,\ \frac{p(\tilde{x})}{q(\tilde{x})}\right); \qquad \text{se rejeitado, reamostrar de } (p-q)_+ = \max(0,\ p-q) \text{ normalizada.}$$

## Alinhamento por Preferências

---

**Reward model** (perda de Bradley-Terry sobre o par preferido $y_w$ e preterido $y_l$):

$$\mathcal{L}_{RM} = -\log \sigma\big(r_\phi(x, y_w) - r_\phi(x, y_l)\big)\,.$$

**RLHF com PPO** (objetivo da política com penalização KL em relação ao modelo de referência):

$$J(\theta) = \mathbb{E}_{x,\,y\sim\pi_\theta}\big[\, r_\phi(x,y) - \beta\,\mathrm{KL}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big)\big]\,.$$

**DPO** (otimiza diretamente sobre os pares de preferência, sem *reward model* explícito):

$$\mathcal{L}_{DPO} = -\log\sigma\left(\beta\left[\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right]\right).$$

## GANs

---

**Loss original do gerador** (saturante), com discriminador $f[\cdot, \phi]$ e gerador $g[\cdot, \boldsymbol{\theta}]$:

$$L[\boldsymbol{\theta}] = \sum_j \ln\Big[\,1 - \sigma\Big(f\big[\,g[z^{(j)}, \boldsymbol{\theta}],\ \phi\,\big]\Big)\Big]\,.$$

**Loss não-saturante** do gerador:

$$L_{\text{ns}}[\boldsymbol{\theta}] = -\sum_j \ln\,\sigma\Big(f\big[\,g[z^{(j)}, \boldsymbol{\theta}],\ \phi\,\big]\Big)\,.$$

**Discriminador ótimo** ($P$ é a densidade dos dados reais e $P^*$ a dos gerados):

$$D^*(x) = \frac{P(x)}{P(x) + P^*(x)}.$$

**Divergência de Jensen-Shannon** e sua decomposição (com $M = \frac{1}{2}(P + P^*)$):

$$D_{JS}[P^* \,\|\, P] = \underbrace{\tfrac{1}{2}\,D_{KL}[P^* \,\|\, M]}_{\text{qualidade}} + \underbrace{\tfrac{1}{2}\,D_{KL}[P \,\|\, M]}_{\text{cobertura}}, \qquad 0 \le D_{JS} \le \ln 2.$$

**WGAN-GP** (*gradient penalty* adicionada à *loss* do crítico, em pontos interpolados $\hat{x}$):

$$\lambda\,\Big(\big\|\nabla_{\hat{x}}\, f[\hat{x}, \phi]\big\|_2 - 1\Big)^2.$$

## Normalizing Flows

---

**Mudança de variáveis** (bijeção $z = f(x)$, $\dim \mathbf{z} = \dim \mathbf{x}$):

$$p_x(\mathbf{x}) = p_z\big(f(\mathbf{x})\big)\,\big|\det J_f(\mathbf{x})\big|, \qquad \log p_x(\mathbf{x}) = \log p_z\big(f(\mathbf{x})\big) + \log\big|\det J_f(\mathbf{x})\big|.$$

**Linear flow:** $\quad f(\mathbf{x}) = \boldsymbol{\theta}\mathbf{x} + \mathbf{b}$.

**Affine coupling** (particiona a entrada em $\mathbf{x}^A, \mathbf{x}^B$; a *coupling network* produz $(\mathbf{s}, \mathbf{t}) = \mathrm{NN}(\mathbf{x}^A)$):

$$\begin{aligned} \text{forward:} \quad & \mathbf{y}^A = \mathbf{x}^A, \qquad \mathbf{y}^B = \mathbf{x}^B \odot \exp(\mathbf{s}) + \mathbf{t}, \\ \text{inversa:} \quad & \mathbf{x}^A = \mathbf{y}^A, \qquad \mathbf{x}^B = \left(\mathbf{y}^B - \mathbf{t}\right) \odot \exp(-\mathbf{s}), \qquad (\mathbf{s}, \mathbf{t}) = \mathrm{NN}(\mathbf{y}^A). \end{aligned}$$

**Log-determinante** da camada *affine coupling*:

$$\log\left|\det J\right| = \sum_i s_i.$$

## Autoencoders Variacionais (VAE)

**KL em forma fechada** entre o posterior $\mathcal{N}(\mu, \Sigma)$ e o *prior* $\mathcal{N}(\mathbf{0}, \mathbf{I})$, com latente de dimensão $D_z$:

$$D_{KL}\left[\mathcal{N}(\mu, \Sigma) \,\middle\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})\right] = \tfrac{1}{2}\left(\mathrm{Tr}[\Sigma] + \mu^\top \mu - D_z - \log\det\Sigma\right).$$

**Reparametrization trick** (com $\epsilon^* \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; para $\Sigma$ diagonal, $\Sigma^{1/2}\epsilon^* = \sigma \odot \epsilon^*$):

$$z^* = \mu + \Sigma^{1/2}\,\epsilon^*.$$

**Termo de reconstrução** (*decoder* gaussiano $p(x \mid z, \boldsymbol{\theta}) = \mathcal{N}_x(f[z, \boldsymbol{\theta}],\, \mathbf{I})$):

$$\tfrac{1}{2}\left\|x - f[z^*, \boldsymbol{\theta}]\right\|^2.$$

**Objetivo** (ELBO; a otimização minimiza −ELBO):

$$-\mathrm{ELBO} \approx \underbrace{\tfrac{1}{2}\left\|x - f[z^*, \boldsymbol{\theta}]\right\|^2}_{\text{reconstrução}} + \underbrace{D_{KL}[q(z \mid x, \phi) \,\|\, p(z)]}_{\text{regularização}}.$$

## Modelos Difusores

**Noise schedule** (produto acumulado a partir de $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_T)$):

$$\alpha_t = \prod_{s=1}^{t} (1 - \beta_s).$$

**Processo forward** (um passo, cadeia de Markov gaussiana, $\epsilon_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$):

$$z_t = \sqrt{1 - \beta_t}\; z_{t-1} + \sqrt{\beta_t}\; \epsilon_{t-1}.$$

**Kernel de difusão** (amostra direta do passo $t$ a partir de $x$ e $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$):

$$z_t = \sqrt{\alpha_t}\; x + \sqrt{1 - \alpha_t}\; \epsilon.$$

**Objetivo de treino** (a U-Net $g_t[\cdot, \boldsymbol{\theta}_t]$ estima o ruído):

$$L_t = \left\| g_t[z_t, \boldsymbol{\theta}_t] - \epsilon \right\|^2.$$

**Amostragem reversa** (passo de *denoising*, com $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ e $\sigma_t$ do *schedule*):

$$z_{t-1} = \frac{1}{\sqrt{1 - \beta_t}}\left(z_t - \frac{\beta_t}{\sqrt{1 - \alpha_t}}\; g_t[z_t, \boldsymbol{\theta}_t]\right) + \sigma_t\, \epsilon.$$

**Classifier-free guidance** (predição condicionada $g_t[z_t, c]$ e sem condição $g_t[z_t, \varnothing]$; peso $w \geq 0$):

$$\tilde{g}_t = (1 + w)\, g_t[z_t, c] - w\, g_t[z_t, \varnothing].$$