Training Objectives for Language Models: Autoregressive, Masked Modeling, and Sampling Strategies

October 25, 2025

1 Autoregressive Language Modeling (Causal LM Loss)

1.1 Likelihood Factorization and Loss

Autoregressive (AR) language models decompose the joint probability of a sequence $x_{1:T}$ via the chain rule:

$$p_\theta(x_{1:T}) = \prod_{t=1}^{T} p_\theta(x_t \mid x_{<t}). \tag{1}$$

Training minimizes the negative log-likelihood, equivalent to token-wise cross-entropy:

$$\mathcal{L}_{\mathrm{AR}}(\theta) = -\sum_{t=1}^{T} \log p_\theta\big(x_t \mid x_{<t}\big) = \sum_{t=1}^{T} \mathrm{CE}\big(\delta_{x_t}, \hat{p}_\theta(\cdot \mid x_{<t})\big), \tag{2}$$

where δ_{x_t} is the one-hot target distribution and $\hat{p}_\theta$ the softmax output. Decoder-only transformers enforce causality through an upper-triangular mask, allowing efficient batched training while aligning precisely with left-to-right generation at inference time.

Teacher forcing feeds ground-truth tokens during training, yielding low-variance gradients. However, at inference the model conditions on its own samples, leading to exposure bias when the generated trajectory drifts into regions unseen during training. Curriculum schedules, data augmentation, and iterative self-refinement help reduce the discrepancy between training and inference distributions.

1.2 Long-Context Modeling

Scaling AR models to long contexts introduces memory and optimization challenges. Practical techniques include:

- **Gradient accumulation:** Emulate large effective batch sizes or context lengths by accumulating gradients across micro-batches before each update.
- **Memory caching:** Architectures such as Transformer-XL and GPT-NeoX reuse hidden states from previous segments to extend the effective receptive field.
- **Advanced positional encodings:** Rotary embeddings (RoPE) and ALiBi biases maintain relative position awareness across extrapolated lengths.

Mixture-of-data training with code, dialogue, and knowledge snippets also improves robustness under long-context prompting by diversifying $p_{\mathrm{data}}(x_{<t})$.

1.3 Regularization and Contrastive Enhancements

Pure language modeling tends to favor fluent yet potentially ungrounded text. Complementary objectives mitigate this tendency:

- **Label smoothing and dropout:** Reduce over-confidence and encourage richer latent representations, especially near rare words.
- **Contrastive terms:** Penalize model agreement with corrupted negatives, shrinking the hypothesis space and reinforcing factual alignment.
- **Curriculum learning:** Start with low-temperature, short-context training and gradually expand complexity to maintain stable gradients.

These additions can be combined with reinforcement learning from human feedback (RLHF) or direct preference optimization (DPO) to align post-training behavior with human expectations.

2 Masked Language Modeling (Masked LM Loss)

2.1 Bidirectional Context Conditioning

Masked language models (MLM) train encoders to reconstruct randomly masked tokens using both left and right context. Given a mask set $\mathcal{M}$, the objective is

$$\mathcal{L}_{\text{MLM}}(\theta) = -\sum_{t \in \mathcal{M}} \log p_\theta(x_t \mid x_{\setminus \mathcal{M}}). \tag{3}$$

MLM provides bidirectional representations ideal for understanding tasks (classification, question answering, named entity recognition), but lacks an inherent generative decoding pathway. Downstream generation usually requires an auxiliary decoder or sophisticated infilling procedure.

2.2 Masking Policies

Mask placement profoundly affects coverage and learning dynamics:

- **Random token masking:** The original BERT recipe replaces 15% of tokens, using an 80/10/10 split among special [MASK], random replacements, and unchanged tokens.
- **Whole-word masking:** Groups wordpieces from the same lexical unit, especially important for languages without explicit whitespace segmentation.
- **Span masking:** SpanBERT and T5 mask contiguous chunks, improving long-range dependency modeling and aligning with text-to-text tasks.
- **Dynamic masking:** Resamples mask locations each epoch so the model observes diverse contexts rather than memorizing fixed patterns.

2.3 Auxiliary Objectives and Joint Training

To bridge the gap between pretraining and downstream tasks, MLM is often combined with complementary objectives:

- **Next sentence prediction (NSP) and sentence order prediction (SOP):** Encourage discourse-level understanding by distinguishing coherent sentence pairs.
- **Replaced token detection (RTD):** ELECTRA trains a discriminator to detect tokens produced by a small generator, yielding efficient representation learning.
- **Instruction-style mixtures:** Frameworks like T5 unify multiple supervised tasks under a span-based generative objective, using the MLM signal as a core component.

Extensions to speech and vision (HuBERT, MAE) demonstrate that masked reconstruction scales to other modalities when paired with appropriate feature extractors and corruption schemes.

3 Tokenization (BPE, SentencePiece, tiktoken)

3.1 Design Principles

Tokenization balances vocabulary size, sequence length, and coverage. Desirable properties include:

- **Completeness:** Every string should be representable without resorting to unknown tokens.
- **Compression efficiency:** Shorter sequences reduce compute, but large vocabularies increase embedding and softmax cost.
- **Multilingual readiness:** Tokenization should gracefully handle diverse scripts, diacritics, and emoji.

Subword units strike a middle ground between character-level robustness and word-level semantic coherence, making them the dominant choice for modern LLMs.

3.2 Byte Pair Encoding (BPE)

BPE iteratively merges the most frequent adjacent token pair (u, v):

1. Initialize the vocabulary with individual characters or bytes.
2. Count frequencies of adjacent token pairs across the corpus.
3. Merge the highest-frequency pair into a new token and replace its occurrences.
4. Repeat until reaching the target vocabulary size.

The deterministic merge table permits reproducible encoding/decoding and adapts well to morphologically rich languages by learning common stems and affixes. For languages like Chinese, byte-level variants avoid the need for external word segmentation.

3.3 SentencePiece and the Unigram LM

SentencePiece operates directly on raw text, emitting subwords using either BPE or a probabilistic unigram language model. The unigram approach maintains a candidate set of subwords, assigning probabilities updated via expectation-maximization:

- Compute sentence likelihoods using forward-backward algorithms under the current subword probabilities.
- Remove low-probability candidates and renormalize until the desired vocabulary size is reached.

Advantages include language-agnostic processing, built-in normalization, and compatibility with special tokens. Models such as T5, mT5, and ALBERT rely on SentencePiece for consistent multilingual coverage.

3.4 tiktoken and Modern Implementations

tiktoken is OpenAI's high-throughput tokenizer for GPT models, designed for large-scale serving:

- **Byte fallback:** Unknown tokens are decomposed into byte sequences, guaranteeing lossless encoding.
- **Optimized data structures:** Trie-based matching and SIMD-optimized kernels yield significant speedups over naive greedy decoding.
- **Model-specific vocabularies:** Packs presets such as `gpt-4` and `cl100k_base`, ensuring training-serving parity.

In production pipelines, tokenizers integrate with normalization, deduplication, and special-role prefixes (e.g., `<|system|>`, `<|assistant|>`) to maintain consistent prompting semantics.

4 Optimization and Sampling Strategies (Teacher Forcing, Top-k, Top-p)

4.1 Teacher Forcing and Exposure Bias

Teacher forcing provides gold-standard context during training, but inference must rely on autoregressive rollouts. The mismatch introduces exposure bias, where early mistakes propagate. Common mitigations include:

- **Scheduled sampling:** Gradually replace a fraction of ground-truth tokens with model predictions, annealing toward free-running generation.
- **Adversarial alignment (Professor Forcing):** Train a discriminator to match hidden-state distributions between teacher-forced and free-running modes.
- **Reinforcement fine-tuning:** Optimize sequence-level rewards (via RLHF, RLAIF, or DPO) to adapt the policy under its own distribution.

4.2 Top-k Sampling

Top-k sampling truncates the distribution to the k highest-probability tokens before renormalization:

$$\mathcal{V}_k = \{x \mid p_\theta(x \mid x_{<t}) \text{ ranks among the top } k\}, \qquad p'(x) = \frac{p_\theta(x \mid x_{<t})}{\sum_{y \in \mathcal{V}_k} p_\theta(y \mid x_{<t})}. \tag{4}$$

Smaller k yields deterministic, high-quality outputs but risks repetitive continuations; larger k offers diversity at the cost of coherence. Temperature scaling rescales logits prior to softmax:

$$p_\tau(x) = \frac{\exp(\log p_\theta(x)/\tau)}{\sum_y \exp(\log p_\theta(y)/\tau)}, \tag{5}$$

with $\tau < 1$ sharpening the distribution and $\tau > 1$ broadening it.

4.3 Top-p (Nucleus) Sampling

Top-p sampling selects the smallest candidate set with cumulative probability at least p:

$$\mathcal{V}_p = \left\{ x : \sum_{y \in \mathcal{V}_p} p_\theta(y \mid x_{<t}) \geq p \right\}, \tag{6}$$

then samples proportionally within $\mathcal{V}_p$. By adapting the truncation threshold to entropy, nucleus sampling maintains fluent output across varying perplexity regimes. Typical values $p \in [0.8, 0.95]$ balance breadth and coherence; hybrid strategies enforce minimum or maximum candidate counts to avoid degenerate cases.

4.4 Repetition Penalties and Diversity Controls

Additional knobs provide finer control over decoding behavior:

- **Repetition penalties:** Scale logits for previously generated tokens by a factor γ to discourage loops.
- **Frequency/presence penalties:** Adjust logits based on occurrence counts (as popularized in OpenAI APIs) to modulate novelty.
- **Contrastive decoding:** Leverage a small auxiliary model to veto low-quality candidates, combining diversity with factual precision.

Evaluation should track not only perplexity but also human-centric metrics such as factuality, harmlessness, and style adherence under the chosen sampling scheme.

5 Practical Considerations

- **Data curation:** Align tokenization, masking, and loss weighting with corpus composition; apply deduplication and filtering to curb memorization and toxicity.
- **Optimization stack:** AdamW, Lion, and decoupled weight decay pair well with gradient clipping, EMA, and mixed precision training to stabilize large-batch regimes.
- **Evaluation and alignment:** Combine automatic metrics (perplexity, BLEU, ROUGE) with human or preference-based assessments tailored to the deployed sampling strategy.

Further Reading

- Bengio et al. “A Neural Probabilistic Language Model.” JMLR, 2003.
- Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” NAACL, 2019.
- Radford et al. “Language Models are Unsupervised Multitask Learners.” OpenAI Technical Report, 2019.
- Holtzman et al. “The Curious Case of Neural Text Degeneration.” ICLR, 2020.
- Raffel et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” JMLR, 2020.