

Estratégias de treinamento em dados ambientais

Cesar de Oliveira F. Silva

Setembro / 2026

Estratégias de treinamento em dados ambientais

- Treinar modelos ambientais envolve desafios específicos:
 - autocorrelação espacial e temporal;
 - desbalanceamento entre classes ou regimes;
 - escassez e heterogeneidade de rótulos.
- O objetivo não é apenas ajustar bem o treino, mas **generalizar para novas regiões, períodos e condições**.

Princípio central

Separar corretamente espaço e tempo no desenho experimental para evitar **data leakage**.

Preparação de dados ambientais

- Harmonização espacial e temporal:
 - reprojeção;
 - alinhamento entre grades e sensores;
 - sincronização temporal entre fontes.
- Padronização das entradas:
 - normalização por banda ou variável;
 - tratamento de escalas distintas;
 - codificação de metadados relevantes.
- Tratamento de faltantes e máscaras:
 - nuvens;
 - falhas de sensor;
 - gaps temporais.

Construção de amostras e janelas

- Em problemas espaciais, é comum trabalhar com:
 - *tiles*;
 - patches;
 - janelas deslizantes.
- Em séries temporais, usa-se *sliding windows* com horizonte de previsão definido.
- O desenho da amostra deve preservar contexto suficiente sem introduzir vazamento entre treino e validação.

Cuidado prático

Janelas próximas no espaço ou no tempo podem parecer amostras distintas, mas conter informação quase redundante.

Balanceamento de classes e eventos raros

- Em dados ambientais, classes raras são frequentes:
 - áreas urbanas pequenas;
 - eventos extremos;
 - classes minoritárias em uso do solo;
 - episódios anômalos de qualidade da água.
- Isso pode levar o modelo a otimizar apenas o comportamento médio.

Estratégias

- oversampling dirigido;
- losses ponderadas;
- focal loss;
- hard example mining.

Métricas sob desbalanceamento

- Métricas globais podem mascarar falhas importantes.
- É recomendável avaliar:
 - desempenho por classe;
 - desempenho por regime;
 - capacidade de detectar eventos raros.

Exemplos

- segmentação: IoU e F1 por classe;
- previsão de eventos: precision-recall, recall em extremos;
- regressão: erro por faixa de magnitude ou por estação.

Escolha e ajuste de hiperparâmetros

- Hiperparâmetros relevantes incluem:
 - learning rate;
 - batch size;
 - profundidade e largura da rede;
 - dropout, weight decay e tamanho de janela.
- O ajuste depende de:
 - escala espacial;
 - horizonte temporal;
 - volume e diversidade dos dados.

Estratégias de busca

grid search, random search e Bayesian optimization.

Learning rate scheduling

- O learning rate controla a dinâmica de convergência.
- Schedulers comuns:
 - step decay;
 - cosine annealing;
 - OneCycleLR.
- Em redes profundas e Transformers, **warm-up** inicial costuma ajudar na estabilidade.

Boa prática

Monitorar simultaneamente curvas de treino e validação para ajustar o regime de otimização e detectar overfitting precoce.

Inicialização e normalização

- A inicialização influencia a estabilidade do gradiente no início do treinamento.
- Opções comuns:
 - Xavier/Glorot;
 - He initialization.
- Normalizações úteis:
 - **BatchNorm**: comum em CNNs com batches grandes;
 - **LayerNorm**: típica em Transformers;
 - **GroupNorm**: útil quando o batch é pequeno.

Efeito prático

Boas escolhas de inicialização e normalização aceleram convergência e reduzem instabilidade numérica.

Regularização prática

- Estratégias clássicas:
 - dropout;
 - weight decay;
 - early stopping;
 - ensembles.
- Em dados ambientais, data augmentation também é central:
 - rotações e flips;
 - jitter espectral;
 - ruído sintético;
 - simulação atmosférica ou perturbações temporais.

Objetivo

Reduzir sobreajuste e aumentar robustez fora do domínio imediato de treino.

Validação adequada em dados ambientais

- A validação deve refletir o cenário real de uso.
- Estratégias principais:
 - **split espacial**: separar regiões geográficas;
 - **split temporal**: rolling forecast ou walk-forward;
 - **split espaço-temporal**: mais próximo do cenário operacional.
- Sempre evitar que amostras próximas no espaço ou no tempo apareçam em treino e teste.

Erro comum

Random split em dados autocorrelacionados quase sempre superestima desempenho.

Transfer learning em imagens de satélite

- Quando há poucos rótulos, pré-treinamento pode acelerar e estabilizar o aprendizado.
- Fontes comuns:
 - ImageNet;
 - datasets remotos como BigEarthNet;
 - encoders auto-supervisionados.
- Em dados multibanda, pode ser necessário adaptar a primeira camada para acomodar novas bandas.

Estratégia comum

Congelar inicialmente o backbone e treinar apenas a cabeça da tarefa específica.

Fine-tuning eficiente

- Após o ajuste inicial da cabeça, é comum refinar o backbone progressivamente.
- Estratégias úteis:
 - **gradual unfreezing**;
 - **discriminative learning rates**;
 - regularização mais forte nas camadas profundas.
- O cuidado principal é o **deslocamento de domínio**:
 - imagens naturais $\neq$ imagens espectrais;
 - uma região $\neq$ outra região;
 - um sensor $\neq$ outro sensor.

Few-shot learning em contextos ambientais

- Situação comum: poucos exemplos rotulados para nova região, novo sensor ou evento raro.
- Abordagens possíveis:
 - meta-learning;
 - prototypical networks;
 - fine-tuning com regularização forte;
 - augmentation direcionada.
- O foco é adaptar rapidamente o modelo sem exigir grande volume de anotação adicional.

Self-supervised learning

- Em muitos domínios ambientais, dados não rotulados são abundantes.
- Tarefas pretexto comuns:
 - masked patch prediction;
 - contraste temporal;
 - reconstrução espectral;
 - previsão como tarefa auxiliar.
- Depois do pré-treinamento, o encoder pode ser ajustado com pequeno conjunto rotulado.

Vantagem

Representações mais robustas e menor dependência de anotação manual extensa.

Experimentos de ablação

- Um bom resultado final não explica, por si só, por que o modelo funciona.
- Ablation studies ajudam a isolar o efeito de cada componente:
 - atenção;
 - skip connections;
 - variáveis exógenas;
 - losses específicas;
 - estratégias de pré-processamento.

Objetivo

Justificar escolhas arquiteturais e identificar o que realmente contribui para o ganho de desempenho.

Análise comparativa de arquiteturas

- Comparações devem usar o mesmo protocolo experimental.
- Não basta comparar apenas a métrica final.
- Também convém avaliar:
 - custo computacional;
 - memória;
 - latência;
 - estabilidade entre seeds;
 - sensibilidade a hiperparâmetros.

Boa prática

Comparar CNNs, Transformers e híbridos sob o mesmo particionamento, mesma métrica e mesmo orçamento computacional.

Reprodutibilidade e rastreabilidade

- Reprodutibilidade exige controle explícito de todo o pipeline.
- Itens mínimos:
 - seeds;
 - versionamento de dados;
 - hiperparâmetros;
 - código e ambiente computacional.
- Ferramentas úteis:
 - MLflow;
 - Weights & Biases;
 - DVC ou versionamento equivalente.

Pipeline

ETL → treinamento → validação → teste → deploy

Ordem prática para montar um experimento

- ① Definir tarefa, unidade de amostra e métrica principal.
- ② Construir split espacial/temporal sem leakage.
- ③ Estabelecer baseline simples e robusto.
- ④ Ajustar modelo principal e hiperparâmetros.
- ⑤ Rodar ablações e comparar custo vs desempenho.

Baselines que não podem faltar

- Persistência / climatologia;
- regressão linear ou árvore;
- CNN ou LSTM simples;
- modelo físico ou híbrido, quando houver.

Princípio

Um modelo sofisticado só é convincente se superar baselines fortes e bem ajustados.

Resumo estratégico

- 1 Desenhe splits espaciais e temporais de forma realista.
- 2 Controle desbalanceamento, regularização e hiperparâmetros com método.
- 3 Use transfer learning e self-supervised learning quando rótulos forem escassos.
- 4 Compare modelos sob protocolo justo e rastreável.
- 5 Avalie não apenas desempenho técnico, mas também custo e utilidade operacional.