

Roadmap para o Desenvolvimento de LLMs Personalizados e Federados para Diagnósticos Médicos

Fase 1: Fundamentação Teórica e Planejamento (Semanas 1-4)

Objetivo: Entender os conceitos de LLMs, Federated Learning, e suas aplicações no diagnóstico médico.

1. Estudo de LLMs e Fine-Tuning

○ Artigos para Ler:

- "Attention Is All You Need" (Vaswani et al., 2017)
- "Language Models Are Few-Shot Learners" (Brown et al., 2020 - GPT-3 paper)
- "An Overview of Fine-Tuning Techniques for Pre-trained Language Models"

○ Ferramentas para Estudo:

- Hugging Face Transformers (documentação e tutoriais).
- Colab ou Kaggle Notebooks para experimentação prática.

2. Estudo de Federated Learning (FL)

○ Artigos para Ler:

- "Federated Learning: Challenges, Methods, and Future Directions" (Li et al., 2020).
- "Advances and Open Problems in Federated Learning" (Kairouz et al., 2019).

○ Ferramentas para Estudo:

- Frameworks como Flower e TensorFlow Federated.
- Google FL Research (explore casos práticos).

3. Estudo sobre Diagnóstico Médico com IA

○ Artigos para Ler:

- "AI in Medical Diagnosis: Opportunities and Challenges"
- "Deep Learning for Healthcare: Review, Opportunities, and Challenges".

○ Ferramentas e Bases de Dados:

- FHIR (Fast Healthcare Interoperability Resources).
- SNOMED CT ou UMLS (para termos médicos).

4. Planejamento do Projeto

- Escolha do LLM Base: e.g., GPT-Neo, Bloom ou Llama.
 - Escolha do framework de FL: e.g., Flower ou TensorFlow Federated.
 - Definição das métricas de sucesso (e.g., precisão do diagnóstico, redução de erro).
-

Fase 2: Configuração do Ambiente e Coleta de Dados (Semanas 5-8)

Objetivo: Preparar o ambiente de desenvolvimento e os dados necessários para treinamento.

1. Configuração do Ambiente

- Configure um ambiente em nuvem (AWS, GCP ou Azure) para treinamentos distribuídos.
- Utilize Docker e Kubernetes para orquestração de containers.
- Ferramentas:
 - Anaconda, PyTorch, Hugging Face Transformers.
 - Flower/TensorFlow Federated para FL.

2. Coleta e Preparação de Dados

- Utilize bases como:
 - **MIMIC-III** (base de dados médicas).
 - **n2c2** (dados anotados para NLP médico).
 - Dados anonimizados do seu app, seguindo a LGPD/HIPAA.
 - Pré-processamento:
 - Normalize textos médicos (SciSpacy).
 - Extraia termos médicos com SNOMED ou UMLS.
-

Fase 3: Fine-Tuning do LLM Base (Semanas 9-12)

Objetivo: Especializar o modelo base para diagnósticos médicos.

1. Fine-Tuning Centralizado (Baseline)

- Realize o fine-tuning inicial de forma centralizada.
- Ferramentas:
 - Hugging Face Transformers (**Trainer** API).
 - PyTorch para customização de treinos.
- **Exemplo:**
 - Utilize o script de fine-tuning do Hugging Face com GPT-Neo e ajuste hiperparâmetros (batch size, learning rate).

2. Validação do Modelo

- Utilize benchmarks como:
 - **n2c2 Challenge** datasets.
 - Métricas: BLEU, F1 Score, precisão no diagnóstico.
-

Fase 4: Implementação do Federated Learning (Semanas 13-16)

Objetivo: Transferir o modelo para um ambiente federado e treinar colaborativamente.

1. Configuração do Federated Learning

- **Ferramentas:**
 - Flower (para orquestração de FL).
 - TensorFlow Federated (para prototipagem).
- **Passos:**
 - Divida os dados entre diferentes simulações de "clientes".
 - Ajuste os algoritmos de agregação (e.g., FedAvg).

2. Treinamento Federado

- Inicialize o modelo base em cada nó.
 - Realize ciclos de treinamento e agregação.
 - Teste o impacto de:
 - Dados heterogêneos.
 - Tamanhos variados de clientes.
-

Fase 5: Personalização Multinível (Semanas 17-20)

Objetivo: Personalizar o modelo para atender a diferentes níveis (regional, institucional, individual).

1. Personalização Regional e Institucional

- Ajuste o modelo federado com dados regionais usando fine-tuning local.
- Métricas: Análise de erros por região.

2. Personalização Individual

- Use dados históricos do paciente como contexto.
 - Técnicas: Few-shot learning ou prompt engineering.
-

Fase 6: Validação e Deploy (Semanas 21-24)

Objetivo: Validar o modelo federado e implementá-lo no app.

1. Validação Extensiva

- Avalie o modelo em cenários reais e simulados.
- Métricas: Tempo de inferência, impacto em diagnósticos.

2. Deploy no App

- Integre o modelo no backend do aplicativo (utilize FastAPI ou Flask).
 - Ferramentas:
 - Docker para criar imagens.
 - Kubernetes para gerenciar serviços escaláveis.
-

Cronograma Consolidado

Fase	Semana	Atividades principais
Fundamentação Teórica	Semanas 1-4	Estudos iniciais, leitura de artigos, planejamento.
Configuração e Coleta	Semanas 5-8	Configuração do ambiente, coleta e preparação de dados.
Fine-Tuning Centralizado	Semanas 9-12	Fine-tuning inicial e validação centralizada.
Federated Learning	Semanas 13-16	Configuração e implementação de FL.
Personalização Multinível	Semanas 17-20	Ajustes regionais, institucionais e individuais.
Validação e Deploy	Semanas 21-24	Validação extensiva e integração no aplicativo.