Aprendizado Federado com AutoKeras e Knowledge Distillation

Bruno H. Meyer[1], **Aurora Pozo**[1], **Michele Nogueira**[2], **Wagner M. Nunan Zola**[1]

[1]Departamento de Informática – Universidade Federal Paraná (UFPR)

[2]Depto. de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)

{bruno, aurora, wagner}@inf.ufpr.br, michele@dcc.ufmg.br

Resumo. *Este artigo apresenta a técnica AFP-KD-AutoML com objetivo de reduzir o tempo de treinamento e execução de modelos para Aprendizado Federado. A técnica usa o conceito Knowledge Distillation para transferir informações entre clientes e servidor e a ferramenta AutoKeras para encontrar arquiteturas de redes neurais artificiais.*

1. Introdução

O Aprendizado Federado (AF) destaca-se por treinar modelos de maneira distribuída e descentralizada. Nessa abordagem, modelos, como Redes Neurais Artificiais (RNA), são treinados em computadores chamados clientes, cada um com seu conjunto de dados. Após treinar com seus dados, cada cliente envia seu modelo treinado para um servidor central que os agrega. Os dados de treinamento não são compartilhados com o servidor central nessa etapa. Posteriormente, o servidor envia um único modelo agregado aos clientes, que o retreinam em seus dados e os reenviam. Essas etapas cíclicas, chamadas de "*rounds*", são executadas até a convergência do modelo ou com a presença de novos dados.

O tempo de treinamento e utilização de modelos provenientes do Aprendizado Federado (AF) configuram desafios em cenários com recursos de processamento limitados. No campo do Aprendizado Federado Personalizado (AFP), a ênfase está em criar modelos específicos para cada cliente, em contraste com a abordagem de um modelo global único. Cada cliente pode ter sua arquitetura de RNA exclusiva, beneficiando-se, no entanto, do modelo global enviado pelo servidor. A busca por hiperparâmetros eficazes em modelos de Aprendizado de Máquina (AM) é desafiadora devido ao extenso espaço de busca, incluindo a otimização da arquitetura de RNA [Garg et al. 2020].

Recentemente, surgiram técnicas como o *Direct Federated NAS* , que busca arquiteturas de RNA em modelos federados através de grafos acíclicos dirigidos. Esta abordagem, apesar de maximizar a eficiência em termos de taxa de predição, negligencia considerações cruciais sobre o tempo de treinamento e execução. Diversas estratégias podem ser adotadas para reduzir o tamanho dos modelos de aprendizado, incluindo o *Auto Machine Learning* (AutoML), uma abordagem para encontrar automaticamente hiperparâmetros adequados, como na busca por arquiteturas de RNA. Uma ferramenta popular de AutoML chamada AutoKeras tem demonstrado eficiência em diversos contextos, seu emprego específico no contexto do AFP, com foco na redução do tempo de treinamento e execução, ainda carece de exploração.

Outra técnica relevante é o *Knowledge Distillation* (KD), onde uma RNA estudante é treinada para reproduzir a saída de uma RNA professora mais complexa. Embora

KD tenha sido explorado no contexto do Aprendizado Federado Personalizado (AFP), nenhuma pesquisa investigou seu uso em conjunto com técnicas de AutoML, além de considerar medidas de restrição de tempo para treinamento e execução [Zhang et al. 2021].

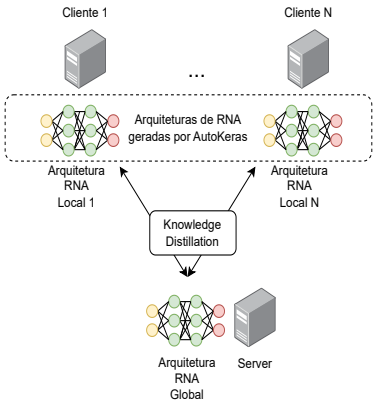
Neste trabalho é proposta uma técnica de AFP chamada AFP-KD-AutoML, onde a transferência de conhecimento entre cliente e servidor é realizada de forma similar ao proposto por [Zhang et al. 2021]. No AFP-KD-AutoML, além do uso de KD, as arquiteturas de RNA dos clientes são construídas considerando limitações de tempo de processamento, o que pode ser obtido por meio de técnicas de AutoML e ferramentas como o AutoKeras.

2. Proposta: AFP-KD-AutoML

A técnica AFP-KD-AutoML tem como principal característica o uso de KD e a busca de arquiteturas usando AutoKeras com restrição de tempo. Os pseudocódigos e um diagrama da proposta são ilustrados na Figura 1. O Algoritmo 1 apresentado descreve um *round* no servidor central, que agrega os modelos locais usando KD para agregar os modelos locais. No Algoritmo 2 são descritas as etapas realizadas pelo cliente em cada *round*, onde o modelo é inicializado considerando restrição de tempo, os pesos são inicializados por KD a partir do modelo global e por fim o modelo é enviado para o servidor. O volume de dados transferidos é equivalente ao número de parâmetros (pesos) das RNAs. O Algoritmo 3 detalha a maneira como o modelo local é treinado e o Algoritmo 4 explica como o modelo local é inicializado.

O AFP-KD-AutoML será avaliado em experimentos com variados valores para restrição de tempo de treinamento e execução dos modelos e comparado com a técnica *Direct Federated NAS* usando simulações, a métrica F1-Score e o tempo de execução. Por fim, espera-se obter um resultado que demonstre a capacidade da proposta AFP-KD-AutoML em gerar modelos com melhor custo-benefício quando se considera tempo de execução e a qualidade dos modelos gerados em AFP.

Figura 1. Algoritmos e diagrama que representam a proposta AFP-KD-AutoML



Algoritmo 1 *Round* no Servidor

```
if primeiroRound( ) then
    Inicializa Modelo
end

ModelosLocais ← EsperaModelosLocais()
ModeloGlobal ← KnowledgeDistillation(ModelosLocais)
EnviaModeloGlobal(ModeloGlobal)
```

Algoritmo 2 *Round* no cliente

```
Entrada: DadosLocais, TempoLimiteTreino, TempoLimiteExecucao
if primeiroRound( ) then
    InicializaModeloLocal(Dados Locais TempoLimiteTreino,
                          TempoLimiteExecucao)
end

ModeloGlobal ← EsperaModeloGlobal()
ModeloLocal ← KnowledgeDistillation(ModeloGlobal)
TreinaModelo(ModeloLocal, DadosLocais,
             TempoLimiteTreino, TempoLimiteExecucao)
EnviaModeloLocal(ModeloLocal)
```

Algoritmo 3 TreinaModelo

```
Entrada: Modelo, DadosLocais, TempoLimiteTreino, TempoLimiteExecucao
t ← TempoExecucao(Modelo, DadosLocais) ;          // Pode variar
if t < TempoLimiteExecucao then
    InicializaModeloLocal(DadosLocais TempoLimiteTreino,
                          TempoLimiteExecucao)
end

ModeloLocal ← MaximizaObjetivo(ModeloLocal, TempoLimiteTreino,
                               DadosLocais)
```

Algoritmo 4 InicializaModeloLocal

```
Entrada: DadosLocais, TempoLimiteTreino, TempoLimiteExecucao
C ← AutoKerasCandidatos() ;    // Hiper-Parâmetros Candidatos
for candidato ∈ C do
    if TempoExecucao(Modelo, DadosLocais) ¿ TempoLimiteExecucao then
        C ← C - {candidato} ;          // Remove o candidato
    end
end
for candidato ∈ C do
    candidato ← MaximizaObjetivo(candidato, TempoLimiteTreino,
                                 DadosLocais)
    Eficiencia(candidato) ← CalculaEficiencia(candidato, DadosLocais)
end
return MelhorEficiencia(C)
```

Referências

Garg, A., Saha, A. K., and Dutta, D. (2020). Direct federated neural architecture search. *arXiv preprint arXiv:2010.06223*.

Zhang, J., Guo, S., Ma, X., Wang, H., Xu, W., and Wu, F. (2021). Parameterized knowledge transfer for personalized federated learning. *Advances in Neural Information Processing Systems*, 34:10092–10104.