

Painel de Proveniência: análise durante o treinamento de Redes Neurais Profundas

Filipe Silva¹, Débora Pina², Liliane Kunstmann², Marta Mattoso²

¹ Engenharia de Computação e Informação, Escola Politécnica, UFRJ

²PESC/COPPE – Universidade Federal do Rio de Janeiro (UFRJ)

dfilipeaugusto@poli.ufrj.br, {dbpina, lneves, marta}@cos.ufrj.br

Abstract. *Training deep neural networks requires the tuning of hyperparameters. This process is costly, and although there are tools for automatically choosing the best hyperparameter setting, the user is responsible for the final decision. To achieve this, it is necessary to analyze the impact of different hyperparameters on metrics such as accuracy and loss. Provenance is a way to represent data derivation relationships, which provide important support in this data analysis. Noting the difficulties for provenance analysis and deep learning, we previously proposed a tool that collects provenance directly from Keras and enables runtime analysis, called Keras-Prov. To enhance its analytical capabilities, we present in this paper an integration of this tool with Elasticsearch and Kibana, creating a Provenance Dashboard for runtime analysis. We present an experimental evaluation of this integration using CNN AlexNet.*

Resumo. *O treinamento de redes neurais profundas requer o ajuste de hiperparâmetros. Este processo é custoso e ainda que existam ferramentas para escolha automática da melhor configuração de hiperparâmetros, o usuário é responsável pela decisão final. Para isso, é necessário analisar o impacto de diferentes hiperparâmetros sobre métricas como acurácia e perda. A proveniência é uma forma de representar as relações de derivação de dados, que fornecem um suporte importante nesta análise de dados. Observando as dificuldades para análises de proveniência e aprendizado profundo, propusemos anteriormente uma ferramenta que coleta proveniência diretamente do Keras e permite análises em tempo de execução, chamada Keras-Prov. Para aperfeiçoar suas capacidades analíticas, apresentamos neste artigo uma integração dessa ferramenta com o Elasticsearch e o Kibana, criando um Painel de Proveniência para análise durante o treinamento. Apresentamos uma avaliação experimental dessa integração usando a CNN AlexNet.*

1. Introdução

O uso de algoritmos de aprendizado de máquina têm se popularizado devido à sua capacidade analítica de dados no apoio à solução de problemas de diversas naturezas. Por outro lado, a própria concepção de modelos de aprendizagem profunda torna seus resultados difíceis de serem explicados ou compreendidos por humanos. [Gunning and Aha 2019].

Considerando que a pesquisa científica é adequada quando documentada com detalhes suficientes para permitir a sua reprodutibilidade e também sua compreensão,

novas demandas são ensejadas para a área de Inteligência Artificial. O dado e suas transformações precisam ser observadas para que a proveniência seja de fato útil [Gil et al. 2019, Pasquier et al. 2017]. Assim, para que os dados de proveniência possam contribuir para a reprodutibilidade e compreensão dos dados gerados no aprendizado de máquina, é necessário que os primeiros sejam analisáveis. É importante que a representação e o armazenamento dos dados de proveniência facilite a análise, por exemplo, por meio de submissão de consultas e por meio de gráficos comparativos.

Apesar de ferramentas de aprendizado de máquina, como o Keras¹, possuírem ambientes de visualização de dados, há limitações quanto à escolha do que pode ser visualizado. Por exemplo, a visualização de dados baseada em resultados de consultas mais complexas sobre os dados de execução, não são simples de serem realizadas. Apesar de diversas ferramentas oferecerem dados da execução em *logs*, observa-se que realizar a sua análise requer bastante esforço por parte do usuário, além de ser propenso a erros. Para realizar adaptações de hiperparâmetros do Keras, é necessário monitorar não apenas a evolução de métricas de treinamento como perda e acurácia, mas seu desempenho computacional ao longo das épocas para uma determinada configuração de hiperparâmetros. Assim, o Keras-Prov [Pina et al. 2020] surge como um sistema que coleta dados de proveniência diretamente ligado ao Keras provendo uma base de dados passível de consultas durante a execução da NN. Além disso, essa base de dados fica persistente, permitindo consultas mesmo após a execução, algo não muito trivial com as exportações de dados dos ambientes associados ao Keras.

O Keras-Prov coleta metadados e dados de proveniência durante a execução gerando uma base de dados usando o SGBD MonetDB. Logo, embora as análises agora estejam disponíveis, por meio de consultas, durante o treinamento, não é trivial analisar o comportamento de métricas apenas por meio de tabelas como resultado de consultas SQL. Faz falta um monitoramento adicional com recursos gráficos e flexibilidade na escolha das visualizações, portanto, a proposta ensejada neste artigo vai ao encontro desta necessidade.

Vale dizer que existem diversas alternativas de visualização de dados de DL²[Kronthaler and Zöllner 2021], entretanto, estas são associadas a dados provenientes de *logs* e desconectadas de uma base de dados de proveniência; assim, para que o usuário associe os dados da visualização com os outros de proveniência, torna-se necessário o desenvolvimento de *softwares* ou processamento de *logs* para ingestão de dados em um SGBD. Na prática, os dados associados à execução do DL ficam separados em arquivos de *logs*, em arquivos de visualização e com acesso por meio de APIs, necessitando de ferramentas para integrá-los a fim de análises envolvendo caminhos de derivação de dados.

Neste artigo apresentamos experimentos com a Rede Neural Convolutiva (CNN) Alexnet [Krizhevsky et al. 2017] que evidenciaram que o apoio visual gráfico, complementar à base de dados de proveniência, permite uma análise evolutiva dos dados ao longo do treinamento da CNN. Visando a aperfeiçoar o processo de análise de dados no Keras-Prov [Pina 2020], este artigo apresenta uma integração da base de dados de proveniência com o ElasticSearch e o Kibana, criando um Painel de Proveniência. O

¹<https://keras.io/>

²<https://www.tensorflow.org/tensorboard>

Elasticsearch e o Kibana foram escolhidos por fazerem parte da *Elastic Stack*³, que traz um conjunto de ferramentas gratuitas e abertas para ingestão, enriquecimento, armazenamento, análise e visualização de dados. Resultados com o Painel de Proveniência sobre a CNN AlexNet evidenciam as vantagens da representação da proveniência para análise de dados, tanto por seguir o padrão W3C PROV [Moreau and Groth 2013], quanto pela riqueza de análises nas consultas e o painel visual.

Este artigo está organizado em outras três seções que dizem respeito, respectivamente, às análises de dados com Keras e Keras-Prov (Seção 2), à proposição de um novo acompanhamento dos dados de proveniência com novas análises sobre a CNN AlexNet (Seção 3) e, finalmente, às considerações finais do trabalho apresentado junto à sua aplicação (Seção 4).

2. Análise de dados com Keras e Keras-Prov

Esta seção tem o intuito de analisar o poder de consultas aos dados de treinamento em CNN utilizando o conjunto de dados *Oxford Flower*, que consiste em dezessete espécies de flores com oitenta imagens para cada classe [Nilsback and Zisserman 2006] com os hiperparâmetros pré-definidos⁴. A instalação do ambiente com o TensorFlow foi feita utilizando uma imagem do *Docker Hub*⁵. Assim, o código original desta aplicação foi executado com 100 épocas utilizando a linguagem Python. A seguir, mostra-se, por meio de tabelas, o comportamento do treinamento dos otimizadores SGD, RMSprop, Adam, Adadelta, Adagrad, Adamax e Nadam utilizando os recursos da API Keras e Keras-Prov.

2.1. API Keras - Realização de Consultas

A utilização isolada da API do Keras para analisar o comportamento dos otimizadores não permite uma análise integrada entre dados de visualização, proveniência e execução ao longo das execuções de cada época. Desta forma, para analisar a perda e acurácia associadas ao término do treinamento com os diversos otimizadores, foi necessário um processamento manual junto ao *log* do Keras para obter tais resultados. A geração dessas análises, como apresentado na Tabela 1, só foi possível após o término de todas as execuções e por meio de um processamento do *log* para extrair dados e inseri-los em um *software* de planilhas externo para a confecção dos gráficos de visualização. Vale dizer que todas as coletas foram realizadas de forma manual, que enseja um enquadramento das métricas importantes que devem ser analisadas.

Tabela 1. Qual a média de tempo de processamento de cada época?

	Adadelta	Adagrad	Adam	Adamax	Nadam	RMSprop	SGD
Média (s)	49.21	48.74	93.52	48.55	50.25	93.81	75.01
Variância	0.19	0.38	128.09	0.27	0.19	126.80	0.35

Avaliando os resultados das tabelas e outros dados do *log*, observa-se a necessidade de analisar mais métricas. Desta forma, para que novas métricas sejam avaliadas, tal como a precisão da validação (*validation accuracy*) e a verificação do processo de *shuffle*, torna-se necessário novas edições na planilha e processamento do *log*.

³<https://www.elastic.co/what-is/elk-stack>

⁴O código original pode ser obtido em <https://github.com/dbpina/data-oxflower>

⁵Disponível em: https://hub.docker.com/r/lilianekunstmann/tensorflow_i/

2.2. Keras-Prov - Realização de Consultas

A abordagem do Keras-Prov incorpora um componente de coleta de armazenamento de dados de proveniência passível de ser consultado ao longo da execução do treinamento. Assim, o Keras-Prov permite combinar o poder de análise do banco de dados relacional colunar MonetDB⁶ - um sistema de gestão de bases de dados de código aberto com um elevado desempenho tanto em inserções, quanto em consultas complexas em relação a grandes bases de dados. Vale dizer que o Keras-Prov conta com uma interface de consultas que gera automaticamente o SQL correspondente. Desta forma, é possível acompanhar e analisar comparativamente os otimizadores durante o treinamento, por exemplo, escolhendo atributos e agregações ao longo das épocas.

A análise das diversas métricas torna-se facilitada ao realizar consultas com o Keras-Prov. Para verificar os melhores valores de acurácia e perda para os otimizadores Adam, SGD e Nadam, pode-se executar a seguinte consulta SQL cujo resultado está na Tabela 2: “*select trainingmodel_task_id as MODEL_ID, max(accuracy) as ACCURACY, min(loss) as LOSS from ds_otrainingmodel where trainingmodel_task_id=1 or trainingmodel_task_id=3 or trainingmodel_task_id=7 group by trainingmodel_task_id;*”.

Além disso, os resultados podem sugerir ao especialista que a rede parou de apresentar melhoria a partir de uma quantidade de épocas, isto é refletido na Tabela 3, obtida a partir seguinte consulta: “*select trainingmodel_task_id, round(accuracy, 3) as accuracy, epoch, round(accuracy - lag(accuracy, 1) OVER (ORDER BY id), 3) as diff_with_previous_acc from ds_otrainingmodel;*”. Esse resultado, aliado ao desempenho computacional (exibido na Tabela 1), pode evidenciar um cenário com diferenças significativas entre os otimizadores. No entanto, para analisar um número maior de épocas, surge a necessidade de complementar os dados numéricos com a visualização desse comportamento.

Tabela 2. Melhores valores de acurácia treino e perda para Adam, SGD e Nadam

Otimizador	Acurácia	Perda
SGD	0.56985	1.15197
Adam	1	0.00025
Nadam	1	0.00022

Tabela 3. Exemplo de acurácia no treinamento

Task ID	Acurácia	Época	Diferença com a acurácia anterior
1	0.063	1	-
1	0.079	2	0.016
1	0.097	3	0.017

Outra vantagem que merece destaque na abordagem Keras-Prov são as funções de agregação que podem ser utilizadas para responder outros tipos de consultas [Pina et al. 2020], por exemplo: “Qual o otimizador demonstrou maior variação no resultado?”, “Qual configuração convergiu mais rápido?” ou “Qual a média de tempo de processamento de cada época?”. Exemplo de consulta SQL: “*select trainingmodel_task_id*

⁶<https://www.monetdb.org>

as modelid, avg(elapsed_time) as average_time, round(max(elapsed_time),3) as max_time, round(min(elapsed_time),3) as min_time from ds_otrainingmodel group by trainingmodel_task_id;”. Por outro lado, uma desvantagem é alteração que o Keras-Prov provoca no core do Keras, o que gera necessidade de atualização de versões.

3. Painel de Proveniência

O Painel de Proveniência consiste em utilizar duas novas ferramentas para que os dados de proveniência do Keras-Prov sejam disponíveis para análise de forma gráfica e interativa. Assim, o Painel de Proveniência, por meio de dados de análise transferidos ao Elasticsearch, provê um mecanismo de consulta e análise rápido e leve de dados que são apresentados graficamente com o Kibana - uma camada de visualização personalizável.

O Painel de Proveniência oferece a escolha de visualização dos principais dados de cada época, por exemplo: número da época, modelo escolhido, tempo decorrido, tempo de início e término, acurácia e perda. Ao enviar esses dados ao Elasticsearch, é possível construir um Painel com gráficos dinâmicos de maneira automática durante a execução, por exemplo: gráfico que apresenta o tempo médio por época; gráficos da acurácia e perda por época; acurácia de treino *versus* acurácia do conjunto de validação; e *etc.*

O Painel⁷ foi desenvolvido em Python e o seu funcionamento principal sugere uma execução periódica para que os dados inseridos no banco de dados MonetDB sejam transferidos, de forma seletiva, ao Elasticsearch, pois o intuito não é ser uma réplica do banco de dados original, mas um motor de busca dos principais dados da aplicação. Portanto, a aplicação do Painel de Proveniência é responsável por utilizar as credenciais do banco de dados MonetDB para que, por meio de uma consulta SQL, possa extrair, a cada intervalo de tempo predefinido, os últimos registros presentes no MonetDB e realizar a ingestão dos mesmos em um motor de busca - Elasticsearch.

```
----- MonetDB - Checking new records -----  
There's nothing to be added!  
Row(s) inserted: 0. Last ID: 600. Last update: 03-01-2021 16:45:58.  
Row(s) inserted: 2. Last ID: 602. Last update: 03-01-2021 16:46:58.
```

Figura 1. Execução da aplicação do Painel de proveniência

O Painel de Proveniência foi avaliado com a mesma AlexNet usado com o Keras-Prov na Seção 2. Cabe observar que, embora os gráficos da Figura 2 possam ser obtidos com APIs de visualização, o Painel permite também visualizar dados de consultas complexas resultantes da base de dados; assim, tem-se flexibilidade no que se deseja visualizar de forma complementar às consultas. Os experimentos do Painel incorporado ao Keras-Prov foram realizados com a suíte de serviços computacionais em nuvem Google Cloud Platform, contendo: 8 vCPUs; 32 GB de memória RAM e 30 GB de SSD. A Figura 2 mostra a disposição gráfica das informações do MonetDB que estão no ElasticSearch. Desta forma, o Kibana, por ser uma plataforma de análise e visualização de dados, facilita o compartilhamento de painéis dinâmicos com variação em tempo de execução, além da criação de gráficos para compreender melhor os dados disponíveis.

Com a visualização provida pelo Painel de Proveniência, pode-se observar anomalias. Na Figura 2, observa-se uma queda repentina nas acurácias de treino e de validação

⁷Disponível em: <https://github.com/dfilipeaugusto/provenancePanel>.

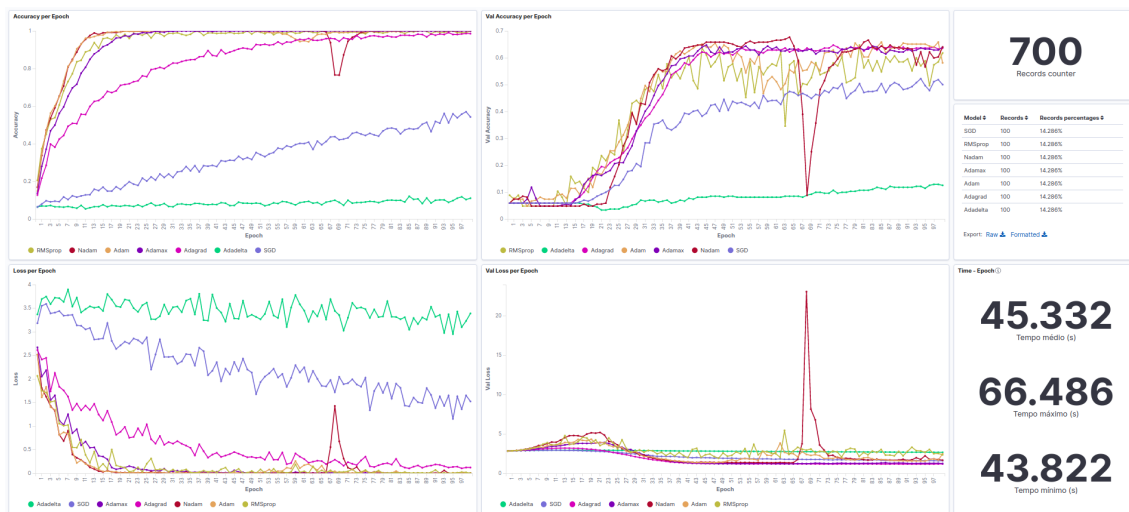


Figura 2. Kibana - Visualização dos dados em tempo de execução

do otimizador Nadam, o que sugere investigação, podendo ser feita por meio da base de dados da Keras-Prov com a utilização de uma simples consulta já apresentada anteriormente⁸ e refletida na Tabela 3. A partir da visualização, também é possível observar se um determinado otimizador apresenta tendências de melhoria ou não, dado importante na tomada de decisão para dar prosseguimento ou interromper a execução. Na Figura 2, observa-se que o otimizador Adadelta não apresenta melhoria em seus valores de acurácia e perda, permitindo ao usuário tomar a decisão de interromper precocemente sua execução, promovendo uma economia de recursos. Ainda é possível comparar gráficos de treino e de validação em busca de indicativos de sobreajuste. A representação desses dados se mostra importante para análises mais ricas que podem ser complementadas através das consultas ao banco de dados.

4. Considerações Finais

As pesquisas recentes mostram que dados de proveniência vêm ganhando proeminência e tende a ser vital para a reprodução, análise e explicação de resultados de redes neurais [Huynh et al. 2018, Namaki et al. 2020]. Neste trabalho, apresenta-se o Painel de Proveniência que visa enriquecer a análise de dados de proveniência em aplicações de CNNs. Desta forma, novos desafios são ensejados e novas perguntas são colocadas: “Os dados coletados são relevantes para as suas análises?”, “Qual combinação sancionou tecnicamente uma decisão?” ou “Será que apenas parte das entradas é relevante?”. A coleta de dados de proveniência em sua extensividade é complexa, pois há vários metadados e dados disponíveis para serem historizados. Como trabalhos futuros, pretende-se explorar: o volume crescente de dados e a sobrecarga ensejada; o grande fluxo de trabalho, ou seja, a quantidade de transformações de dados ao longo do caminho; a gerência da granularidade dos dados de proveniência.

⁸Consulta SQL: “select trainingmodel_task_id as modelid, avg(elapsed_time) as average_time, round(max(elapsed_time),3) as max_time, round(min(elapsed_time),3) as min_time from ds_otrainingmodel group by trainingmodel_task_id;”

Agradecimentos

Trabalho realizado com apoio do CNPq, FAPERJ e da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

- Gil, Y., Honaker, J., Gupta, S., Ma, Y., D’Orazio, V., Garijo, D., Gadewar, S., Yang, Q., and Jahanshad, N. (2019). Towards human-guided machine learning. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 614–624.
- Gunning, D. and Aha, D. (2019). Darpa’s explainable artificial intelligence (xai) program. *AI Magazine*, 40(2):44–58.
- Huynh, T. D., Ebden, M., Fischer, J., Roberts, S., and Moreau, L. (2018). Provenance network analytics. *Data Mining and Knowledge Discovery*, 32(3):708–735.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90.
- Kronthaler, F. and Zöllner, S. (2021). *Data Analysis with RStudio*. Springer.
- Moreau, L. and Groth, P. (2013). Provenance: an introduction to prov. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 3(4):1–129.
- Namaki, M. H., Floratou, A., Psallidas, F., Krishnan, S., Agrawal, A., Wu, Y., Zhu, Y., and Weimer, M. (2020). Vamsa: Automated provenance tracking in data science scripts. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1542–1551.
- Nilsback, M. . and Zisserman, A. (2006). A visual vocabulary for flower classification. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, volume 2, pages 1447–1454.
- Pasquier, T., Lau, M. K., Trisovic, A., Boose, E. R., Couturier, B., Crosas, M., Ellison, A. M., Gibson, V., Jones, C. R., and Seltzer, M. (2017). If these data could talk. *Scientific Data*, 4(1):170114.
- Pina, D. (2020). Captura de dados de proveniência para apoiar a análise de hiperparâmetros em redes de aprendizado profundo. Dissertação de mestrado PESC/COPPE/UFRJ.
- Pina, D., Kunstmann, L., Oliveira, D., Valduriez, P., and Mattoso, M. (2020). Uma abordagem para coleta e análise de dados de configurações em redes neurais profundas. In *Anais do XXXV SBBD*, pages 187–192.