UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

JOSÉ IGNÁCIO MORSCH SCHMID

SEU MODELO ESTÁ EM PERIGO?

UM ESTUDO DE CASO SOBRE REPLICAÇÃO DE MODELOS USANDO DESTILAÃO DE CONHECIMENTO COM DADOS FORA DO ESCOPO

CAMPINA GRANDE - PB

2020

JOSÉ IGNÁCIO MORSCH SCHMID

SEU MODELO ESTÁ EM PERIGO?

UM ESTUDO DE CASO SOBRE REPLICAÇÃO DE MODELOS USANDO DESTILAÃO DE CONHECIMENTO COM DADOS FORA DO ESCOPO

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Professor Dr. Leandro Balby Marinho.

CAMPINA GRANDE - PB

2020

S348s Schmid, José Ignácio Morsch.
Automação para o projeto Laserterapia da Universidade Federal de Campina Grande. / José Ignácio Morsch Schmid. – 2020.

10 f.

Orientador: Prof. Dr. Leandro Balby Marinho.
Trabalho de Conclusão de Curso - Artigo (Curso de Bacharelado em Ciência da Computação) - Universidade Federal de Campina Grande; Centro de Engenharia Elétrica e Informática.

1. Aprendizagem de máquina. 2. Destilação de conhecimento. 3. Redes neurais. 4. Classificação de imagens. 5. Acurácia. 6. Algoritmo de aprendizagem de máquina. I. Marinho, Leandro Balby. II. Título.

CDU:004(045)

Elaboração da Ficha Catalográfica:

Johnny Rodrigues Barbosa
Bibliotecário-Documentalista
CRB-15/626

JOSÉ IGNÁCIO MORSCH SCHMID

SEU MODELO ESTÁ EM PERIGO?

UM ESTUDO DE CASO SOBRE REPLICAÇÃO DE MODELOS USANDO DESTILAÃO DE CONHECIMENTO COM DADOS FORA DO ESCOPO

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Professor Dr. Leandro Balby Marinho
Orientador – UASC/CEEI/UFCG

Professor Dr. Maxwell Guimarães de Oliveira
Examinador – UASC/CEEI/UFCG

Professor Dr. Tiago Lima Massoni
Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 2020.

CAMPINA GRANDE - PB

Seu modelo está em perigo? Um estudo de caso sobre replicação de modelos usando Destilação do Conhecimento com dados fora do escopo

José Ignácio Morsch Schmid
ignacio.schmid@gmail.com
Universidade Federal de Campina Grande
Campina Grande, Paraiba

RESUMO

Destilação do conhecimento é uma técnica que permite transferir o conhecimento de um modelo de aprendizagem de máquina já treinado para um outro modelo utilizando apenas suas saídas. Sabendo que a replicação do comportamento de um modelo utilizando apenas esses dados é factível, torna-se relevante considerar o fator de proteção da propriedade intelectual quando oferecendo as predições de um modelo em um ambiente em que os usuários possam fazer um grande número de acessos. Neste trabalho iremos fazer o uso da destilação do conhecimento como meio para replicar um modelo convolucional de classificação, tendo acesso apenas a suas predições em um dado fora do escopo original de classificação de modo a avaliar se existe um risco do modelo ser roubado uma vez que alguém tenha amplo acesso a ele.

KEYWORDS

Machine Learning, Destilação do conhecimento, Segurança

1 INTRODUÇÃO

Redes Neurais são algoritmos de aprendizado de máquina que modelam funções complexas que dificilmente poderiam ser descritas explicitamente. Esses algoritmos têm redefinido o estado da arte em tarefas de diversos domínios, como Visão Computacional[14] e Processamento de Linguagem Natural[1]. Os modelos de redes neurais têm se tornado cada vez mais comuns no dia a dia, estando presentes na internet, nos celulares e até em ambientes médicos para a detecção de doenças[12]. Conforme seu uso cresce, surge a necessidade do estudo de técnicas a fim de melhorar a eficiência e segurança de tais modelos.

A destilação do conhecimento, em inglês *Knowledge Distillation*, é uma técnica proposta por Hinton[3] com o objetivo inicial de comprimir um modelo, ou seja, passar o conhecimento de um modelo com uma arquitetura complexa e robusta para outro com uma arquitetura mais simples a fim de melhorar seu tempo de resposta. Esse processo é capaz de transferir o conhecimento de um ou mais modelos de redes neurais já treinadas, chamadas de modelo(s) professor(es), para um outro, chamado de modelo aluno, de modo que o aluno tenha as mesmas saídas que o professor para as mesmas entradas, fazendo com que o aluno seja uma réplica funcional do professor. Em seu trabalho, Hinton demonstra que é possível fazer com que um aluno mais simples que o professor consiga ter respostas muito semelhantes ao professor usando os dados de treino para a destilação.

Sabendo que a replicação do comportamento de um modelo apenas a partir de entradas arbitrárias e suas respectivas saídas é factível, torna-se relevante considerar o fator de proteção da propriedade intelectual quando oferecendo as predições de um modelo em um ambiente em que os usuários possam fazer um grande número de acessos. Neste trabalho consideramos o cenário em que um atacante tem acesso a um modelo de classificação de imagens de forma caixa-preta, isto é, tem acesso apenas as suas entradas e saídas, porem sem ter conhecimento das classes preditas. Nessas condições um atacante poderia obter acesso a esse serviço/API e buscar replicar o modelo, cenário esse especialmente perigoso em aplicações onde os dados são sensíveis como contextos médicos, policiais e bancários, uma vez que as previsões dos modelos poderiam expor de alguma forma esses dados. Um exemplo seria um serviço de detecção de pessoas com antecedentes de crimes graves, visto que a previsão desse modelo poderia revelar se uma pessoa já cometeu ou não tais crimes em sua vida baseado, apenas, em sua imagem. Situações semelhantes também podem ocorrer quando uma empresa oferece as predições de um modelo como um serviço, uma vez que um usuário mal intencionado poderia replicá-lo e deixar de utilizá-lo.

Neste trabalho iremos fazer o uso da destilação do conhecimento como meio para replicar um modelo convolucional de classificação de imagens utilizando apenas dados de fora de seu escopo original, isto é, dados de um contexto diferente dos usados no treinamento do professor e que não condizem com a finalidade do modelo professor. O principal objetivo deste trabalho é verificar se é factível replicar um modelo tratando-o como uma função de caixa preta, avaliando, dessa forma, se existe um risco real dele ser replicado por alguém que tenha acesso ilimitado a ele, mesmo que não saiba sua função. Usaremos os dados de fora do escopo, pois tratando-se de uma situação de modelo caixa preta, não sabemos os dados originais usados para o seu treinamento.

Iremos utilizar a acurácia como principal métrica ao analisar o aluno e o professor, e iremos compará-los utilizando os resultados da predição nos *datasets* usados para o treinamento do professor. Também iremos realizar uma análise do processo de destilação com o objetivo de obter uma melhor compreensão deste.

Esse documento está organizado como se segue: na Seção 2 apresentamos a Fundamentação Teórica, na Seção 3, os Trabalhos Relacionados, na Seção 4, a Metodologia de desenvolvimento do experimento, na Seção 5, os Resultados, e, por fim, na Seção 6, a Conclusão e Trabalhos Futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Redes convolucionais

Uma rede convolucional é um tipo de rede neural utilizada para o processamento de imagens. Neste trabalho, todos os experimentos serão realizados com redes convolucionais com o fim de classificar imagens. Esse tipo de rede foi escolhido por ser um dos tipos usados por Hinton[3] em seu trabalho.

Esse tipo de rede realiza um mapeamento de um conjunto de *pixels* para a classificação de um objeto. Ela faz isso através de uma série de filtros responsáveis por extrair características da imagem de maneira incremental. A primeira camada, por exemplo, seria responsável por extrair características mais simples, como a detecção de bordas. Já na segunda, as características detectadas na primeira seriam usadas para identificar características mais complexas, como cantos e contornos. A terceira poderia identificar formas a partir desses contornos, como por exemplo um círculo. Após extrair essas informações da imagem, a rede utiliza essas características para gerar um conjunto de valores que representam as diversas classes. Quanto maior o valor de uma classe, maior a chance da imagem ser da respectiva classe de acordo com o modelo.

2.2 Destilação do conhecimento

O processo de destilação descrito por Hinton[3] em seu artigo consiste em fazer o modelo aluno entender os dados baseado na percepção que o modelo professor tem desses mesmos dados. Ao utilizar a predição do professor como rótulo do aluno, obtemos muito mais informações do que usando os rótulos originais.

Para entendermos melhor a intuição por trás desse processo, podemos pensar num modelo que classifica imagens contendo números nos números correspondentes. O rótulo original de um desenho de um número "6" traz apenas a informação de seu valor, isto é, a classe correspondente ao número "6", enquanto que a predição de um modelo para o mesmo desenho pode trazer muito mais informações. Além da alta probabilidade de ser um "6", a predição pode incluir uma probabilidade razoável de ser um "8" devido ao círculo na parte inferior do número. Assim, o modelo aluno vai entender as similaridades entre os dois números.

O processo de destilação faz uso de uma equação de erro que utiliza o rótulo original do dado, chamada, neste trabalho, de verdade absoluta, junto à predição do professor, que será chamada de rótulo suave.

A equação de erro usada é a seguinte:

$$ERRO(A_T, P_T, T_Y) = EC(A_T, P_T) + \alpha * EC(A_T, T_Y) \quad (1)$$

Em que A_T e P_T são as predições dos modelos aluno e professor, respectivamente, para um *dataset* T, T_Y é a verdade absoluta do *dataset* T, EC a função de entropia cruzada(Apêndice B) e α um peso que, segundo Hinton, tem como objetivo reduzir o impacto da verdade absoluta no calculo do erro.

O algoritmo da destilação é o seguinte:

(1) O modelo professor P é treinado com os dados T. Sendo P_T a predição de P para os dados T.
(2) Destile o modelo P para o modelo A utilizando o dataset T:
  (a) Faça a predição de P, para os os dados T, para obter o valor de P_T.
  (b) Faça a predição de A, para os dados T para obter o valor de A_T.
  (c) Treine o modelo de forma a minimizar o $ERRO(A_T, P_T, T_Y)$.
(3) Avalie seu modelo A utilizando os dados de T.

É importante ressaltar que nos experimentos do trabalho de Hinton[3] todos os modelos foram redes neurais convolucionais de classificação e a função de saída usada foi a *softmax*. Esta função recebe o conjunto de valores resultado da rede convolucional e retorna uma distribuição de probabilidades indicando a chance do dado ser de uma determinada categoria. No artigo, o autor conseguiu resultados melhores durante a destilação ao usar valores maiores que o valor padrão de uma variável chamada de temperatura (Apêndice A), parâmetro da função *softmax*. Esse parâmetro funciona como uma maneira de suavizar a distribuição das probabilidades fazendo com que as baixas probabilidades da previsão fiquem mais altas e tenham mais impacto na destilação.

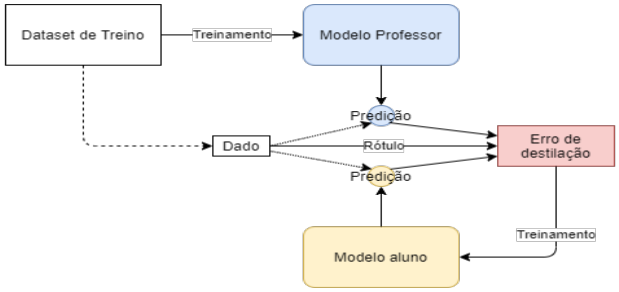

Figura 1: Esquema representando o processo de destilação usado por Hinton

2.3 VGG

O VGG é uma arquitetura de rede convolucional proposta por Simonyan[13] que foi construída com o objetivo de reconhecimento e classificação de imagens na base de dados ImageNet[10]. Neste trabalho, usaremos uma de suas versões, a VGG16, como base para o modelo professor. Esta escolha foi feita devido ao fato da arquitetura já ser usada como referencia na literatura[4][2] e por outros também já a terem utilizado como base[8] para outras arquiteturas.

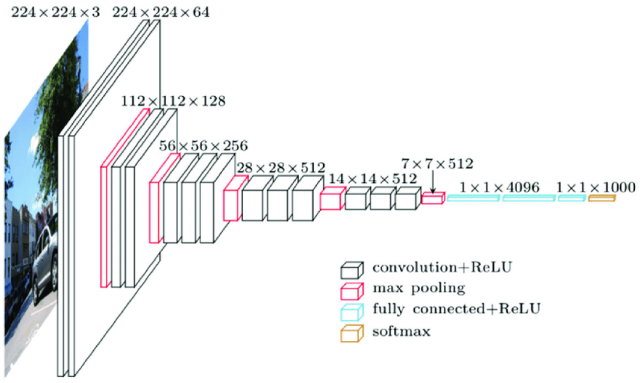

Figura 2: Arquitetura da VGG16

3 TRABALHOS RELACIONADOS

Na literatura revisada, vários trabalhos fizeram uso de dados fora do escopo junto ao processo destilação para diversos objetivos.

Kulkarni[7] obteve resultados positivos em seus experimentos ao utilizar dados fora do escopo como forma de aumentar os dados de treino no processo de destilação, além de ter bons resultados com os mesmos dados quando os de treino não estavam disponíveis.

Menghani e Ravi [9] propõem um processo de refinamento de modelos que utiliza a refinação com dados fora do escopo como uma das etapas e obtêm resultados ainda melhores que o professor.

Ambos os trabalhos demonstram ser possível realizar a destilação usando dados fora do escopo. A principal diferença do nosso trabalho em relação aos supracitados está nas arquiteturas utilizadas. Em ambos os trabalhos, tanto para professor quanto para aluno, as arquiteturas utilizadas foram pequenas. As maiores usadas possuíam na faixa de 3 milhões de parâmetros, enquanto no presente trabalho usamos uma arquitetura baseada na VGG16 que é mais robusta e possui 14.8 milhões de parâmetros.

4 METODOLOGIA

4.1 Arquitetura

Para a arquitetura do modelo professor, foi utilizado como base o VGG16[13], realizando mudanças nas camadas densas para adaptar o modelo para as classes presentes no *dataset* que será usado.

Já para o modelo do aluno utilizamos duas arquiteturas diferentes. Uma delas é a própria arquitetura do professor e a outra baseada no MobileNetV2[11], também com as mudanças similares para adequá-la ao *dataset* usado.

A escolha por reutilizar a arquitetura VGG16 para destilação foi feita por simular uma situação de ataque em que, além das informações estabelecidas anteriormente na Seção 1, também se conhece a arquitetura. Já a da MobileNetV2 foi escolhida por ser comumente utilizada para destilação, por ter menos parâmetros, por ser mais eficiente quanto ao uso de recursos e, ao mesmo tempo, obter boa performance em tarefas como ImageNet.

4.2 Datasets

Foram utilizados dois *datasets* diferentes, o CIFAR-10 e o CIFAR-100[6]. Cada um possui, respectivamente, 10 e 100 classes. O primeiro foi usado como *dataset* de treinamento para o professor, definindo, consequentemente, as classes que serão preditas pelo modelo. Já o segundo é o *dataset* de destilação. Ambos possuem 60.000 imagens que são divididas em duas partes, a parte de treino, a qual possui 50.000 imagens, e a de teste, que possui 10.000. É importante ressaltar que não há nenhuma classe que esteja em ambos *datasets* e, em razão disso, o CIFAR-100 foi considerado de um contexto diferente do CIFAR-10.

Como nenhuma validação ou teste são realizados usando o *dataset* de destilação, utilizamos as suas duas partes como *datasets* separados, sendo o *dataset* de destilação grande os dados da parte de treino do CIFAR-100, e o *dataset* de destilação pequeno os dados da parte de teste.

Todos os dados utilizados nos experimentos foram pré-processados utilizando a normalização das cores tendo como referência o *dataset* ImageNet[10]. Tal metodologia foi adotada pois, ao comparar os resultados com e sem o pré-processamento, os resultados com ele indicaram uma maior acurácia. A escolha pela utilização desse *dataset* como referência foi realizada em razão de ambas as arquiteturas terem sido treinadas, inicialmente, para a classificação do ImageNet, concluindo-se, portanto, que as duas conseguem extrair as características das imagens de maneira satisfatória.

4.3 Treinamento

O treinamento do modelo professor foi realizado em duas etapas, a primeira de *transfer learning* e a segunda de *fine tuning*.

O processo de *transfer learning* envolve a reutilização de parte de uma rede treinada para uma tarefa em uma outra tarefa. Camadas de uma rede são congeladas de modo a impedir o treinamento das mesmas, enquanto outras ficam livres para serem treinadas. Neste processo, as partes congeladas são normalmente as aproveitadas do modelo que está sendo reutilizado. Esse método permite reaproveitar características já aprendidas pela rede base, a qual normalmente é treinada em uma extensa base de dados.

Já o processo de *fine tuning*, por sua vez, descongela parte das camadas congeladas e realiza uma nova etapa de treinamento com taxa de aprendizado baixa, refinando, assim, as características reaproveitadas, afim de que elas sejam mais significantes para o novo propósito final da rede.

Para o *transfer learning*, utilizamos a parte convolucional do VGG pré-treinada no ImageNet[10], a qual foi congelada, para o treinamento do resto do modelo. Foram utilizados os pesos disponibilizados pelo Keras[5]. O modelo rodou durante 10 épocas, com o otimizador Adam com uma taxa de aprendizado de 10^{-3}.

Para o processo de *fine tuning* foram descongeladas as duas últimas camadas convolucionais e treinadas por 7 épocas, também com otimizador Adam, mas com a taxa de aprendizado reduzida para 10^{-5}.

A cada época de treino, tanto no processo de *transfer learning* quanto no de *fine tuning*, algumas das imagens randomicamente escolhidas foram alteradas como forma de aumentar os dados. As alterações realizadas foram a inversão horizontal e a mudança no brilho. Os resultados com o aumento de dados obtiveram melhores valores de acurácia do que sem referido aumento. O tamanho do lote em ambas as etapas foi de 32.

Todos os hiperparâmetros foram selecionados de maneira empírica através de várias execuções. Foram testados valores de épocas entre 5 e 50, tendo sido o melhor escolhido visando a maximização da acurácia e a redução do *overfitting*. A função de erro utilizada no treinamento foi a entropia cruzada.

4.4 Destilação

Para a destilação, nos baseamos no artigo de Hinton[3], com duas mudanças relacionadas ao cenário que estamos propondo. Na primeira mudança consideramos o modelo como uma caixa preta, então não temos acesso a mudanças no parâmetro de temperatura (Apêndice A) e, por isso, usaremos o valor padrão. A segunda é em relação à presença dos dados originais. Na equação 1, a verdade absoluta é utilizada, entretanto estamos assumindo que não temos acesso aos dados de treino. Sendo assim, a função será apenas a entropia cruzada da predição do aluno com o rótulo suave. Tais rótulos suaves serão obtidos utilizando o *dataset* de destilação.

A intuição por trás da destilação com dados fora do escopo é semelhante à intuição da seção 2.2, ou seja, um classificador de imagens de números, entretanto, ao invés de utilizarmos imagens de número para a destilação, utilizamos letras. Ao tentarmos prever a letra ”b” por exemplo, podemos obter uma alta probabilidade de ser um ”6”, devido à similaridade da forma, e uma probabilidade menor de ser um ”8” pela mesma similaridade do círculo inferior. O mesmo acontece ao prevermos um ”C”, o qual pode ter uma probabilidade razoável de ser um ”0”, em razão da forma de semi-círculo, e também de ser um ”6”, por ser similar à haste do número. Assim, o modelo aluno vai aprendendo características do número ”6” sem nunca ter visto o mesmo, de maneira que, quando ele eventualmente tentar prever um ”6”, ele será capaz de identificar as características aprendidas e prever corretamente.

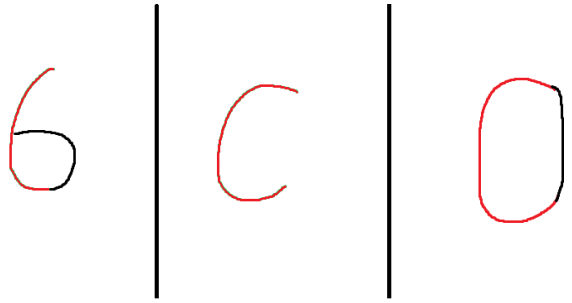

Figura 3: Similaridades entre número ”6”, letra ”C” e número ”0”, respectivamente

A função de erro utilizada foi a seguinte:

$$ERRO(A_X, P_X) = EC(A_X, P_X) \tag{2}$$

Sendo A_X e P_X as predições do modelo A e P, respectivamente, para um *dataset* X e EC a função de entropia cruzada.

O processo de destilação também é parecido ao descrito anteriormente, sendo as duas diferenças a função de erro a ser minimizada e o fato de termos dois *datasets* diferentes. O algoritmo é da seguinte forma:

(1) O modelo professor P é treinado com os dados do *dataset* T.
(2) Destile o modelo P para o modelo A utilizando o dataset D:
  (a) Faça a predição de P, para os os dados D, para obter o valor de P_D.
  (b) Faça a predição de A, para os dados D, para obter o valor de A_D.
  (c) Treine o modelo de forma a minimizar o $ERRO(A_D, P_D)$.
(3) Avalie seu modelo A utilizando os dados de T.

Sendo P e A os modelos professor e aluno e T e D os *datasets* de treino e de destilação respectivamente.

Em todos os experimentos o modelo foi destilado por 100 épocas. Foi também utilizado o otimizador Adam com uma taxa de aprendizado de 10^{-3}. O tamanho do lote utilizado foi de 32.

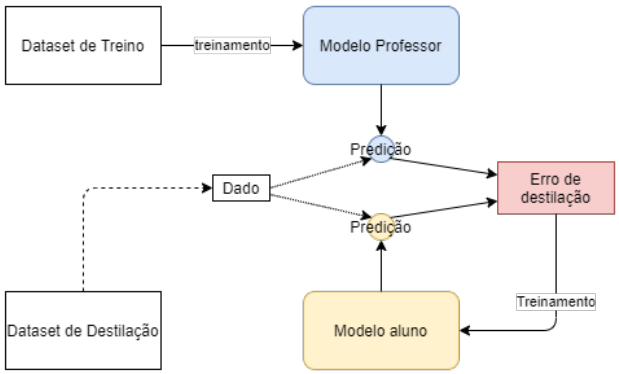

Figura 4: Esquema representando o processo de destilação com dados fora do escopo

5 RESULTADOS

Tabela 1: Métricas do modelo Professor

Dataset	Acurácia	Erro
Treino	0.86	0.40
Teste	0.81	0.59

Os resultados do modelo professor no *dataset* de teste serão utilizados como base para a análise dos experimentos a seguir. Os *datasets* de treino e teste são as respectivas partes do CIFAR-10[6] usados no treinamento e avaliação do modelo professor, seus resultados podem ser vistos na Tabela 1.

Em todas as medições, os valores foram arredondados para duas casas decimais.

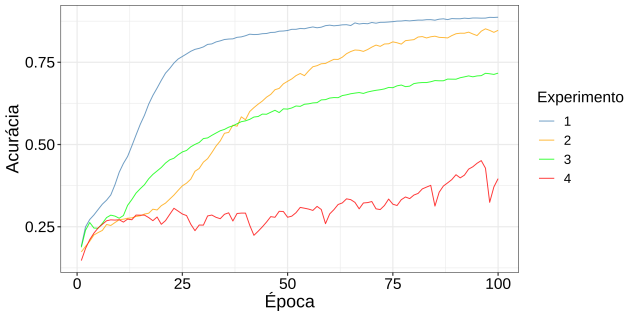

Figura 5: Acurácia de destilação ao longo das épocas

Nas sessões abaixo, o número do experimento está seguido da arquitetura usada pelo aluno e do *dataset* utilizado na destilação.

5.1 Experimento 1: Arquitetura VGG, Dataset grande

Esse experimento obteve resultados inesperados com valores de erro muito grandes em algumas épocas arbitrárias. Inicialmente, o

resultado parece ser fruto da escolha da semente aleatória ao inicializar o modelo aluno, entretanto, outras repetições do experimento obtiveram resultados semelhantes. Esse comportamento aconteceu em ambos *datasets* e algumas vezes acontecia em apenas um deles, como na época 75, em que isso aconteceu apenas no *dataset* de teste, e na época 100, que aconteceu apenas no *dataset* de treino. O valor obtido da acurácia foi razoável. Mesmo em situações em que o erro estava muito alto ele não parece ser afetado. Na época 100, sua melhor época, o modelo aluno deste experimento conseguiu uma acurácia equivalente a 85% da acurácia do modelo professor.(Tabela 2)

Tabela 2: Desempenho do experimento 1 em diversas épocas da destilação

Dataset	Métrica	25	50	75	100
Treino	Acurácia	0.65	0.68	0.68	0.69
	Erro	1.3×10^6	0.92	0.93	1.08×10^8
Teste	Acurácia	0.65	0.67	0.67	0.69
	Erro	4.5×10^6	0.94	142×10^3	0.92

5.2 Experimento 2: Arquitetura VGG, Dataset pequeno

A destilação com o *dataset* pequeno conseguiu uma acurácia muito baixa em relação ao *dataset* grande em sua melhor época, atingindo uma acurácia de 48%, equivalente a 59% da acurácia alcançada pelo modelo professor. Podemos observar também que, apesar da acurácia estar baixa, os resultados para ambos *datasets* estão muito similares, sendo este um indicativo que o modelo aluno consegue generalizar bem. (Tabela 3)

Tabela 3: Desempenho do experimento 2 na época 100

Dataset	Acurácia	Erro
Treino	0.48	21.04
Teste	0.48	21.17

5.3 Experimento 3: Arquitetura MobileNetV2, Dataset grande

Os resultados deste experimento foram um pouco inferiores aos do experimento 1 com o mesmo conjunto de dados. Entretanto, os valores do erro não cresceram tanto quanto o Experimento 1. A acurácia atingida foi de aproximadamente 78% em relação da do professor.(Tabela 4)

5.4 Experimento 4: Arquitetura MobileNetV2, Dataset pequeno

O experimento 4 teve resultados muito inferiores ao experimento 2. Esse modelo conseguiu atingir 42% da acurácia alcançada pelo modelo professor. Assim como o experimento 2, os valores de treino e de teste ficaram muito similares.(Tabela 5)

Tabela 4: Desempenho do experimento 3 em diversas épocas da destilação

Dataset	Métrica	25	50	75	100
Treino	Acurácia	0.51	0.56	0.61	0.63
	Erro	1.76	2.50	1.11	1.07
Teste	Acurácia	0.50	0.55	0.60	0.62
	Erro	1.83	2.55	1.13	1.09

Tabela 5: Desempenho do experimento 4 na época 100

Dataset	Acurácia	Erro
Treino	0.34	1.72
Teste	0.34	1.72

6 CONCLUSÃO E TRABALHOS FUTUROS

Pudemos ver que os experimentos com o *dataset* grande obtiveram resultados muito melhores do que com o pequeno. Isso é um forte indicativo de que assim como o processo de treinamento usual, o processo de destilação também depende de uma grande quantidade de dados e que mais dados levam a melhores resultados. Neste trabalho, tamanhos maiores de *datasets* não foram avaliados devido a limitações na infraestrutura disponível, de modo que o processamento de tais *datasets* seria inviável.

Conseguimos mostrar que é possível sim realizar a destilação do conhecimento com dados fora do escopo, porém é necessário considerar a taxonomia(Apêndice C) dos *datasets* utilizados. Os dados fora do escopo foram todos de um único *dataset*, contendo imagens de taxonomia semelhante ao do problema original, animais e veículos, de modo que não podemos afirmar se o sucesso do processo de destilação foi resultado da escolha deste dataset de destilação específico para o objetivo final específico ou não.

É interessante também estudar a possibilidade de um *dataset* de destilação com taxonomia mais variada, de maneira que seja possível medir o impacto da variabilidade de taxonomias de um dataset no processo de destilação.

Em relação à arquitetura do modelo aluno, o modelo baseado em VGG obteve a melhor acurácia, entretanto não é possível afirmar que isso foi uma vantagem do VGG, por ser igual ao modelo professor ou uma desvantagem do MobileNetV2 por ser uma arquitetura menor. Para afirmar isso seriam necessários mais experimentos com outras arquiteturas diferentes de tamanho semelhante.

Na Figura 5, podemos ver a acurácia de destilação, isto é, a acurácia do aluno em relação às predições do professor nos dados fora do escopo, no *dataset* de destilação, através das diversas épocas. Podemos ver que após a época 75 o aumento da acurácia de destilação é muito pequeno, entretanto é existente, contínuo e termina ocasionando numa melhora na acurácia do objetivo final, conforme podemos observar nas Tabelas 2 e 4. Isso é um forte indicativo de que aumentar o número de épocas poderia melhorar ainda mais o resultado.

Esse fato chama atenção, pois em um processo de treinamento tradicional de um modelo, treinar por muitas épocas quando o

ganho de acurácia entre elas é ínfimo termina ocasionando em *overfitting*, entretanto, por estarmos utilizando o dataset de destilação, esse "*overfitting* de destilação" parece fazer com que os resultados se aproximem ainda mais do professor sem impactar de maneira negativa nos resultados do objetivo final. Embora possamos afirmar que uma melhoria na acurácia de destilação cause uma melhoria na acurácia final, um valor alto da acurácia de destilação não acarreta num valor alto de acurácia final. Isso pode ser percebido ao comparar a acurácia de destilação dos experimentos 1 e 2, que embora tenham valores próximos, os resultados finais são muito diferentes.

Podemos observar, ainda, a destilação como um possível regularizador, dado que a diferença entre a acurácia de treino e de teste dos experimentos, nas épocas analisadas, não passou de 0.0, enquanto que no modelo professor essa diferença foi de 0.05.

A partir dos resultados apresentados, vimos que é possível realizar destilação do conhecimento com dados fora do escopo, uma vez que a acurácia dos alunos foi melhor que um modelo aleatório, que com 10 classes obteria um resultado de 0.10 de acurácia, de modo que podemos afirmar que parte do conhecimento do professor foi de fato transferido para o aluno.

Entretanto, não podemos afirmar que os modelos obtidos são usáveis, uma vez que a usabilidade de um modelo com os valores de acurácia obtidos vai depender muito do contexto do problema. Então, mesmo que ele seja um modelo usável, nossos resultados não nos permitem afirmar que um modelo está em perigo de ser replicado hoje. Entretanto, também não podemos afirmar o contrário, visto que os resultados obtidos neste trabalho são reflexo direto do conjunto de técnicas e dados utilizados neste estudo de caso. Ainda assim, nós encontramos diversos pontos que podem vir a melhorar os resultados de futuros processos de destilação, de modo que a replicação de modelos usando destilação do conhecimento com dados fora do escopo parece ser algo factível que pode vir a ser comprovado em novos trabalhos.

REFERÊNCIAS

[1] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. 2014. Deep Speech: Scaling up end-to-end speech recognition. arXiv:cs.CL/1412.5567

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:cs.CV/1512.03385

[3] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. arXiv:stat.ML/1503.02531

[4] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:cs.CV/1704.04861

[5] KERAS [n. d.]. Keras. Retrieved Nov 05, 2020 from https://keras.io

[6] Alex Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images.

[7] Mandar Kulkarni, Kalpesh Patil, and Shirish Karande. 2017. Knowledge distillation using unlabeled mismatched images. arXiv:cs.CV/1703.07131

[8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single Shot MultiBox Detector. *Lecture Notes in Computer Science* (2016), 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

[9] Gaurav Menghani and Sujith Ravi. 2019. Learning from a Teacher using Unlabeled Data. arXiv:cs.LG/1911.05275

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. arXiv:cs.CV/1409.0575

[11] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2019. MobileNetV2: Inverted Residuals and Linear Bottlenecks. arXiv:cs.CV/1801.04381

[12] Muhammad Sharif, Muhammad Attique Khan, Muhammad Rashid, Mussarat Yasmin, Farhat Afza, and Urcun John Tanik. 2019. Deep CNN and geometric features-based gastrointestinal tract diseases detection and classification from wireless capsule endoscopy images. *Journal of Experimental & Theoretical Artificial Intelligence* (2019), 1–23.

[13] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:cs.CV/1409.1556

[14] Xudong Sun, Pengcheng Wu, and Steven C. H. Hoi. 2017. Face Detection using Deep Learning: An Improved Faster RCNN Approach. arXiv:cs.CV/1701.08289

A TEMPERATURA

A temperatura é um parâmetro da função softmax, que é a seguinte:

$$softmax(x_i, T) = \frac{\exp(x_i/T)}{\sum_j \exp(x_j/T))} \tag{3}$$

Onde T é a temperatura e x é o conjunto de valores em que desejamos aplicar a função. Para alterarmos a temperatura a partir da *softmax*, seria necessário inverter a função e ter acesso ao valor de temperatura usado. Como a maioria dos modelos utiliza o valor padrão, 1, podemos desconsiderar esse fator.

Sendo S a função *softmax*, x_i um item qualquer do conjunto de valores e j o total de valores no conjunto, podemos escrevê-la como:

$$S(x_i) = \exp(x_i)/(\sum_j \exp x_j) \tag{4}$$

Aplicando o logaritmo neperiano em ambos os lados:

$$\ln(S(x_i)) = \ln(\exp(x_i)/\sum_j \exp(x_j)) \tag{5}$$

$$\ln(S(x_i)) = \ln(\exp(x_i)) - \ln(\sum_j \exp(x_j)) \tag{6}$$

$$\ln(S(x_i)) = x_i - \ln(\sum_j \exp(x_j)) \tag{7}$$

Isolando x_i:

$$x_i = \ln(S(x_i)) + \ln(\sum_j \exp(x_j)) \tag{8}$$

O segundo termo da equação é constante para um determinado conjunto x e por isso pode ser escrito como C. Dessa forma, podemos escrever a função como:

$$x_i = \ln(S(x_i)) + C \tag{9}$$

De modo que o resultado da inversa sempre seria função de uma constante qualquer, não sendo possível inverter a *softmax*.

B ENTROPIA CRUZADA

A equação da entropia cruzada é a seguinte:

$$EntropiaCrusada(y', y) := -\sum_i y'_i \ln(y_i) \tag{10}$$

Onde y e y' são duas distribuições de probabilidades a serem comparadas.

C TAXONOMIA

Como vimos na Imagem 3, as características das classes originais são interpretadas a partir da imagem sendo destilada. Quando falamos de variabilidade taxonomia do *dataset* neste trabalho, nos referimos à variabilidade das características que são interpretadas nas previsões. Dentro do exemplo de classificação de números usando letras para a destilação, podemos pensar nas letras "B" e "W" como partes de taxonomias diferentes, uma vez que elas possuem formas muito distintas. No processo de destilação, imagens de mesma taxonomia têm grandes chances de terem previsões similares feitas pelos professor. Em um *dataset* com uma mesma taxonomia, ou uma baixa variabilidade de taxonomia, é provável que o aluno aprenda bem as características que são comuns em tais taxonomias e aprenda mal as que não são comuns.

Ainda no contexto de destilação de um classificador de números utilizando letras, podemos pensar que ao usar um *dataset* de destilação que contém em grande parte letras mais arredondadas como "B","C","O" e "S", obteríamos resultados melhores em números arredondados como "5","8","0","6". O mesmo poderia ocorrer com letras com ângulos mais agudos como "A","V","W" e "L", que obteriam melhores resultados em números com ângulos similares como "1","4' e "7". Assim, uma variedade de taxonomias deve resultar em uma melhor habilidade de generalização do modelo aluno.

Embora seja difícil estabelecer os limites de uma taxonomia, identificar que imagens são de diferentes taxonomias pode ser fácil. Neste trabalho, utilizamos o CIFAR-100[6] para destilação. Nele existem algumas taxonomias que são facilmente distinguíveis, como, por exemplo, meios de transporte, peixes, mamíferos e veículos.