Particionamento de Redes Neurais Profundas com Saídas Antecipadas

Roberto G. Pacheco, Rodrigo de Souza Couto *

[1]Universidade Federal do Rio de Janeiro - PEE/COPPE/GTA

{pacheco, rodrigo}@gta.ufrj.br

Resumo. *Este trabalho aborda o problema de particionamento de Redes Neurais Profundas (Deep Neural Networks - DNNs) com saídas antecipadas, para classificação de imagens. Essa solução objetiva reduzir o tempo de inferência, considerando que a borda executa parte da DNN e a nuvem executa a outra parte. O problema consiste em escolher qual camada divide a DNN em duas partes. A dissertação modela o particionamento como um problema de caminho mais curto em um grafo, sendo resolvido em tempo polinomial. O modelo é usado como base para propor o sistema POPEX (Partitioning OPtimization for deep neural networks with Early eXits). Além disso, analisa-se o impacto da distorção da imagem e da calibração da DNN no particionamento. Este trabalho mostra que essa solução proposta reduz o tempo de inferência e aprimora as decisões de offloading à nuvem.*

1. Motivação e Descrição dos Problemas

Aplicações de visão computacional empregam Redes Neurais Profundas (*Deep Neural Networks* - DNNs) para executar diversas tarefas, como classificação de imagens e detecção de objetos. Contudo, os modelos de DNNs demandam alto poder computacional, o que pode não estar disponível em dispositivos móveis, impedindo diversas aplicações. Para contornar esse problema, a dissertação combina DNNs com saídas antecipadas e particionamento de DNNs.

2. Trabalhos Relacionados

Esta seção apresenta trabalhos relacionados a DNNs com saídas antecipadas e particionamento de DNNs.

DNNs com saídas antecipadas: são DNNs que permitem concluir a inferência de determinadas amostras antecipadamente nas camadas intermediárias, caso a confiança de classificação atinja um limiar. BranchyNet [Teerapittayanon et al., 2016] apresenta proposta de DNNs com saídas antecipadas para reduzir o tempo de inferência. Outros trabalhos utilizam saídas antecipadas para reduzir o consumo de energia [Bolukbasi et al., 2017] ou até mesmo selecionar o tamanho adequado do modelo, como o Edgent [Li et al., 2019]. Este trabalho utiliza a proposta da BranchyNet para implementar DNNs com saídas antecipadas. Contudo, nenhum dos trabalhos anteriores propõe particionar o modelo de saídas antecipadas.

*O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001. O trabalho também foi financiado pelo CNPq, pela FAPERJ com os auxílios E-26/203.211/2017, E-26/201.833/2019 e E-26/211.144/2019, e pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), auxílio no. 2015/24494-8.

Particionamento de DNNs: consiste em determinar, por meio de um problema de otimização, a camada de particionamento que é responsável por dividir a DNN em duas partes. Uma parte executa na borda e a outra na nuvem. Neurosurgeon [Kang et al., 2017] e DADS [Hu et al., 2019] (*Dynamic Adaptive DNN Surgery*) propõem métodos de particionamento ótimo para reduzir o tempo de inferência. O primeiro realiza uma busca exaustiva, enquanto o segundo modela o problema como um grafo. A partir do modelo, o DADS trata o problema de particionamento como um problema de corte mínimo. Contudo, os trabalhos anteriores não abordam o particionamento de DNNs com saídas antecipadas. [Teerapittayanon et al., 2017] propõem um particionamento estático da BranchyNet ao longo de uma arquitetura composta de dispositivo final, borda e nuvem. Por outro lado, a dissertação propõe um sistema de particionamento dinâmico, conforme as condições da infraestrutura de rede. Além disso, diferente dos trabalhos citados, este aborda o problema de calibração de DNNs com saídas antecipadas, mostrando que a calibração aprimora as decisões de *offloading* realizadas pelo dispositivo em borda.

3. Particionamento de DNNs com saídas Antecipadas

A Figura 1 ilustra o particionamento de DNNs com saídas antecipadas entre um dispositivo em borda e a nuvem. Nesse cenário, ao receber uma imagem de entrada, o dispositivo em borda a processa até a camada de particionamento v_s. Após essa camada, as demais são processadas remotamente na nuvem. Dessa forma, o problema de particionamento consiste em determinar, por meio de um problema de otimização, uma camada de particionamento v_s. Este trabalho determina um particionamento ótimo que minimize o tempo de inferência, ou seja, o tempo necessário para classificar uma dada imagem.

As DNNs com saídas antecipadas permitem concluir a inferência de uma imagem nas camadas intermediárias com base em um critério de confiança. Essa classificação antecipada permite reduzir o tempo de inferência. Para isso, insere-se ramos laterais nas camadas intermediárias da DNN, como mostrado na Figura 1. Ao receber uma imagem de entrada, o dispositivo da borda executa a DNN. A DNN processa a imagem, camada a camada, até alcançar o i-ésimo ramo lateral. Esse ramo lateral calcula a entropia de classificação e verifica se esse valor é menor que um dado limiar. Este trabalho utiliza a métrica de entropia, que representa a incerteza da classificação. Nesse caso, a inferência é concluída e o i-ésimo ramo lateral classifica a amostra. Caso contrário, a imagem é processada pelas próximas camadas até o próximo ramo lateral, no qual o procedimento anterior se repete. Caso nenhum ramo lateral alocado na borda seja capaz de classificar a amostra, ela é enviada à nuvem.

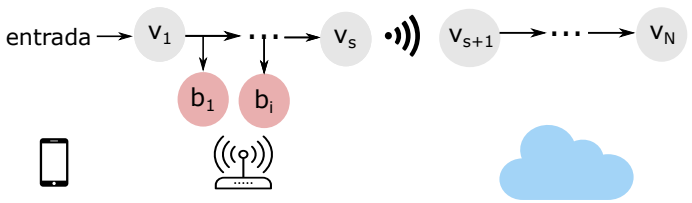


Figura 1. Ilustração de uma DNN com saídas antecipadas usando particionamento entre borda e nuvem.

O particionamento de DNNs com saídas antecipadas depende do tempo de processamento na borda e na nuvem, do tempo de comunicação e das probabilidades de

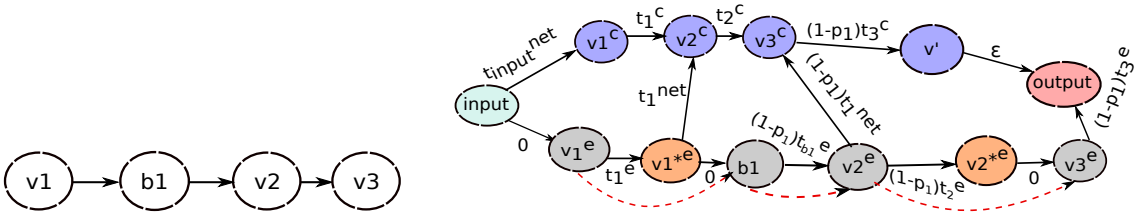
classificação em cada um dos ramos laterais na DNN. No caso de particionamento, o tempo de comunicação corresponde ao tempo necessário para enviar os dados de saída da camada de particionamento v_s da borda à nuvem. Com objetivo de particionar DNNs com saídas antecipadas, a dissertação propõe o sistema POPEX (*Partitioning OPtimization for deep neural networks with Early eXits*)[1], cujo objetivo é minimizar o tempo de inferência. Esta dissertação também mostra que há um compromisso entre acurácia e tempo de inferência em DNNs com saídas antecipadas. Tal compromisso ocorre, pois, à medida que mais amostras são classificadas nos ramos laterais, o tempo de inferência diminui. Contudo, isso acarreta uma queda indesejável de acurácia. Portanto, propõe-se encontrar o particionamento que equilibre o compromisso entre acurácia e tempo de inferência. A arquitetura do sistema POPEX divide-se em três componentes: *Background*, *Update Box* e *Decision Maker*. O *Background* extrai os tempos de processamento de cada camada executada na borda e na nuvem, além do tamanho em *bytes* dos dados de saída de cada uma das camadas da BranchyNet. Essa tarefa é executada apenas durante a inicialização do sistema. Por fim, o *Background* modela como um grafo a arquitetura da BranchyNet fornecida pela aplicação. A Figura 2(a) ilustra o grafo de uma DNN com saídas antecipadas com três camadas e um ramo lateral, denotado pelo vértice b_1. Assim, modela-se o problema de particionamento de DNNs como um particionamento de grafos, de modo que cada camada corresponde aos vértices do grafo e a comunicação entre camadas corresponde às arestas. Após extrair os parâmetros, o *Background* fornece-os ao componente *Decision Maker*.

O *Update Box* monitora as condições de taxa de envio da infraestrutura de rede. Em seguida, calcula-se o tempo de comunicação para enviar os dados de saída de cada camada à nuvem, usando a taxa de envio e o tamanho desses dados. O *Update Box* fornece, periodicamente, o tempo de comunicação das camadas ao *Decision Maker*. Dessa forma, o *Update Box* torna o sistema dinâmico, já que a taxa de envio depende das condições da infraestrutura de rede.

A partir dos parâmetros extraídos pelos componentes anteriores, o *Decision Maker* executa a tarefa de determinar o particionamento ótimo e enviar à nuvem essa decisão e os dados de saída da camada de particionamento. Para isso, o *Decision Maker* constrói o grafo de particionamento, que permite associar um atraso (processamento na borda, na nuvem e o atraso de comunicação) a cada uma das arestas desse grafo de particionamento. A Figura 2(b) ilustra o grafo de particionamento construído, a partir do grafo da Figura 2(a). Na Figura 2(b), os vértices cinza e laranja são processados na borda, enquanto os vértices azuis são processados na nuvem. Note que, na Figura 2(b), cada caminho entre os vértices *input* e *output* corresponde a uma decisão de particionamento diferente. A partir da construção do grafo de particionamento, mostra-se que o problema de particionamento pode ser modelado como o problema de caminho mais curto. Portanto, é possível aplicar o algoritmo de Dijkstra. Consequentemente, resolve-se o particionamento em tempo polinomial com complexidade computacional de $\mathcal{O}(m + n\log(n))$, sendo m e n o número de arestas e vértices do grafo de particionamento, respectivamente.

Este trabalho avalia, por meio de simulação, os impactos da probabilidade de classificação nos ramos laterais e das capacidades de processamento entre borda e nuvem no tempo de inferência obtido pelo problema de otimização. Para implementar uma BranchyNet, usa-se uma DNN como base e insere-se ramos laterais em sua estrutura.

[1] https://github.com/pachecobeto95/POPEX

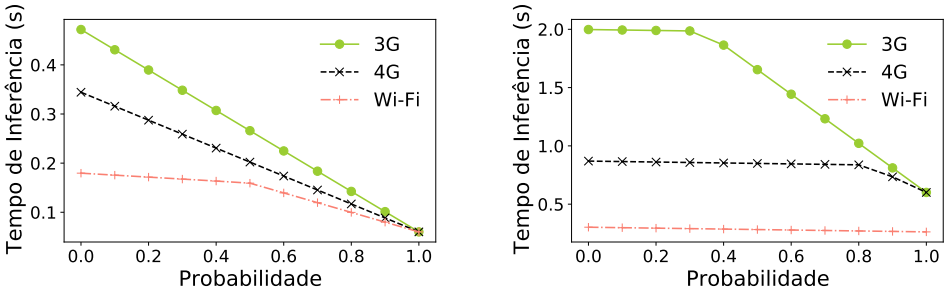


(a) Grafo extraído pelo *Background*.

(b) Grafo de particionamento.

Figura 2. Construção do grafo de particionamento

Este trabalho utiliza como base a AlexNet, uma DNN com cinco camadas convolucionais, seguida de três totalmente conectadas. Em seguida, introduz-se um ramo lateral após a primeira camada convolucional. A posição do ramo principal é escolhida para evitar processamento desnecessário na borda. Para obter o tempo de processamento de cada camada na nuvem, utiliza-se o *hardware* disponível no Google Colaboratory. Em relação à borda, define-se o tempo de processamento na borda da i-ésima camada, denotado por t_i^e, como sendo $t_i^e = \gamma \cdot t_i^c$, no qual $\gamma \in \mathbb{N}_{>1}$ é um fator multiplicativo que indica a razão entre tempo de processamento na borda e na nuvem. A análise considera que o dispositivo de borda utiliza uma determinada tecnologia sem-fio para enviar dados à nuvem, assumindo que o gargalo da comunicação é a rede de acesso da borda. O experimento usa taxas de envio fixas de $1,1$; $5,5$ e $18,8$ Mbps, obtidas de [Hu et al., 2019], correspondendo às tecnologias sem-fio 3G, 4G e Wi-Fi.



(a) Borda com alto poder de processamento ($\gamma = 10$).

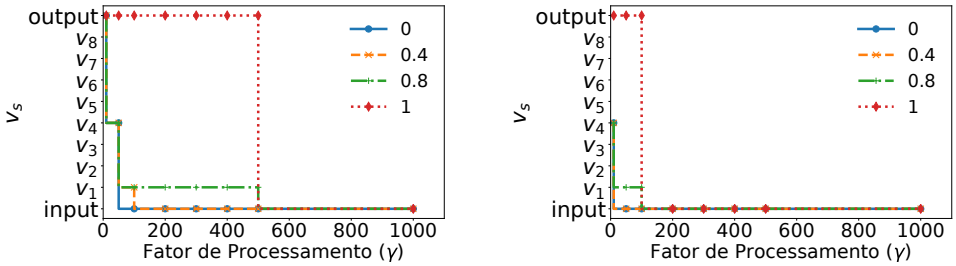
(b) Borda com baixo poder de processamento ($\gamma = 100$).

Figura 3. Tempo de inferência em função da probabilidade de classificação de uma amostra para diferentes tecnologias sem-fio e fatores de processamento γ.

A Figura 3 mostra como a probabilidade de classificação nos ramos laterais impacta o tempo de inferência para dois fatores de processamento γ: 10 e 100. Além disso, os resultados mostram que tecnologias de comunicação sem fio com taxas de envio menores são mais afetadas pela probabilidade de classificação nos ramos. Por exemplo, os resultados usando taxa de envio de 1,1 Mbps (3G), na Figura 3(a), mostram que o tempo de inferência reduz em 87,27%, comparando a probabilidade igual a zero com o caso em que a probabilidade é 1, pois quando a probabilidade de classificação nos ramos laterais é baixa, o problema de particionamento escolhe processar parte das camadas na nuvem. Na Figura 3(b), o eixo y permanece constante em um dado intervalo de probabilidade, pois o problema de otimização escolhe processar todas as camadas na nuvem, já que uma grande parcela das amostras não é classificada no ramo lateral. Por outro lado, à medida que mais

amostras são classificadas no ramo lateral, o tempo de inferência decresce e se torna mais vantajoso utilizar a borda. Usando o mesmo cenário descrito anteriormente, analisa-se a escolha da camada de particionamento v_s em função do fator de processamento γ com diferentes probabilidades de classificação. As Figuras 4(a) e 4(b) mostram a camada escolhida em função dos fatores de processamento, usando taxas de envio correspondente às tecnologias sem fio 3G e 4G, respectivamente. Quando a camada de particionamento é *input*, isso significa que todas as camadas da DNN são processadas na nuvem. Por outro lado, quando a camada de particionamento é *output*, todas as camadas da DNN são processadas na borda. Nessas figuras, cada curva representa uma dada probabilidade de classificar a amostra no ramo lateral. A Figura 4(a) mostra que, à medida que o fator de processamento aumenta, a camada de particionamento escolhida tende a se aproximar da camada de entrada. Por exemplo, assumindo uma probabilidade $p = 0,8$, a Figura 4(a) mostra que, a partir de $\gamma = 500$, a camada de particionamento muda de v_1 a *input*. Isso significa que, a partir desse γ, o processamento é realizado exclusivamente na nuvem. Esse comportamento é esperado, já que a borda, com baixo poder de processamento, faz o problema escolher processar as camadas na nuvem. Quando se compara a Figura 4(b) com a Figura 4(a), nota-se que, para a rede com taxa de envio de 5,5 Mbps, o problema começa a escolher a estratégia de processamento apenas na nuvem (isto é, *output*) para um fator de processamento mais baixo (ou seja, para um maior poder de processamento na borda). Isso ratifica a tendência observada na Figura 3, na qual, para altas taxas de envio, o problema possui uma maior tendência a escolher a estratégia de processamento exclusivamente na nuvem. Por fim, a Figura 4 também confirma o comportamento da Figura 3, em que a probabilidade influencia a escolha da camada de particionamento e, portanto, impacta o tempo de inferência.

Os resultados deste trabalho validam a proposta de combinar particionamento de DNNs com saídas antecipadas como uma solução para reduzir o tempo de inferência. Portanto, essa estratégia é uma alternativa para viabilizar aplicações sensíveis à latência.



(a) Análise com taxa de envio de 1,1 Mbps. (b) Análise com taxa de envio de 5,5 Mbps.

Figura 4. Camada de Particionamento para diferentes fatores de processamento.

4. Impacto da distorção da imagem no particionamento de DNNs com saídas antecipadas

Os resultados anteriores mostram que a probabilidade de classificação nos ramos laterais altera a decisão de particionamento em DNNs com saídas antecipadas. Esta seção, por sua vez, demonstra que a qualidade da imagem impacta a probabilidade de classificação e, consequentemente, a decisão de particionamento. Para isso, implementa-se uma B-AlexNet para classificar imagens e insere-se distorção *blur* nas imagens variando

a qualidade delas. A B-AlexNet é treinada usando um *dataset* composto de imagens de cães e gatos em diferentes ambientes sem nenhuma distorção. Após treiná-la, aplica-se diferentes níveis de *blur* gaussiano a um lote com 48 imagens, empregando filtros gaussianos com dimensões 5, 15 e 65 para representar imagens com distorção baixa, intermediária e alta. A Figura 5 mostra a probabilidade de classificar uma amostra no ramo em função do limiar de entropia. Esse resultado mostra que imagens com maior distorção apresentam menores probabilidades de serem classificadas antecipadamente nos ramos laterais. Logo, o nível de distorção da imagem afeta as probabilidades de classificação nos ramos laterais e, consequentemente, o particionamento.

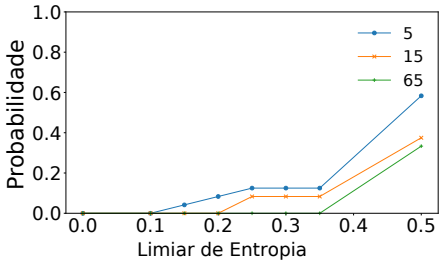


Figura 5. Probabilidade de classificação antecipada com vários níveis de *blur*.

5. Calibração de DNNs com saídas antecipadas

As DNNs com saídas antecipadas precisam prover uma métrica precisa de incerteza para decidir se a inferência pode ser concluída antecipadamente nos ramos laterais. Contudo, Guo *et al.* [Guo et al., 2017] mostram que as DNNs são descalibradas, isto é, superestimam a sua capacidade de inferência, provendo classificações superconfiantes. Guo *et al.* não lida com DNNs com saídas antecipadas. Este trabalho aborda o problema de DNNs com saídas antecipadas descalibradas, que podem decidir de forma equivocada classificar amostras nos ramos laterais, reduzindo a sua acurácia. Para isso, avalia-se o número de amostras classificadas nos ramos e sua acurácia em uma DNN com saídas antecipadas descalibrada e calibrada usando o método *Temperature Scaling* [Guo et al., 2017].

A análise avalia como o método *Temperature Scaling* altera o número de amostras classificadas no primeiro ramo lateral. A Figura 6 mostra a porcentagem das amostras classificadas no primeiro ramo lateral de acordo com a configuração do limiar, antes e depois da calibração. Conforme a figura, a calibração reduz a porcentagem de amostras classificadas no ramo lateral, quando comparado com o caso em que não se aplica a calibração. Esse resultado ocorre pois o ramo superestima a sua capacidade de inferência, classificando mais amostras do que deveria.

Para confirmar que os ramos superestimam sua capacidade de inferência, avalia-se o impacto do método de calibração na acurácia. A Figura 7 mostra a acurácia em função do limiar de entropia usando uma B-AlexNet com apenas um ramo lateral. Em geral, a Figura 7 mostra que uma DNN com um ramo calibrado atinge uma acurácia maior, quando comparado com o ramo lateral descalibrado. Portanto, aplicar uma etapa de calibração nos ramos laterais pode aumentar a acurácia das DNNs com saídas antecipadas. Isso ocorre pois o ramo lateral descalibrado superestima sua confiança em relação à sua capacidade de inferir amostras antecipadamente. Logo, esse ramo classifica mais amostras, como já

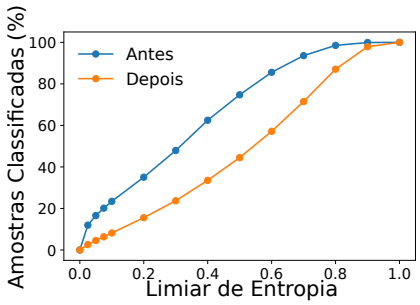


Figura 6. Porcentagem de amostras classificadas no primeiro ramo lateral.

mostrado na Figura 6. Isso faz com que as DNNs classifiquem amostras pouco confiáveis nos ramos laterais, reduzindo a acurácia da DNNs com saídas antecipadas.

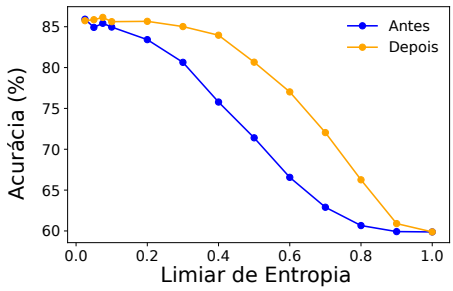


Figura 7. Tempo de Processamento na nuvem de cada camada da AlexNet.

6. Produções Científicas e Tecnológicas

As publicações diretas da dissertação foram os artigos listados a seguir:

1. **R. G. Pacheco** e R. S. Couto, ”Introduzindo a qualidade da imagem como uma nova condição de particionamento de DNN na borda”, em Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (**SBRC**), 2020, p. 1–14. Qualis A4.
2. **R. G. Pacheco** e R. S. Couto, “Inference time optimization using Branchynet partitioning”, em IEEE Symposium on Computers and Communications (**ISCC**), 2020, pp. 1–6. Qualis A3.

Além disso, os resultados da análise de calibração de DNNs com saídas antecipadas foram aprimorados, com a inclusão de métricas para avaliar o impacto da calibração em um cenário de particionamento, culminando na seguinte publicação:

- **R. G. Pacheco**, R. S. Couto e O. Simeone, “Calibration-aided Edge Inference Offloading via Adaptive Model Partitioning of Deep Neural Networks”, em IEEE International Conference on Communications (**ICC**), 2021, aceito para publicação. Qualis A1.

Por fim, a implementação da dissertação resultou no seguinte programa de computador registrado:

1. POPEX (*Partitioning OPtimization for deep neural networks with Early eXits*), registrado no INPI sob o número BR512020001239-6

7. Conclusões

A dissertação propôs o POPEX, um sistema que determina o particionamento ótimo de DNNs com saídas antecipadas para minimizar o tempo de inferência. Para tal, o POPEX modela o problema de particionamento em um problema de caminho mais curto, resolvendo-o em tempo polinomial com algoritmo Dijkstra. Os resultados mostram que o processamento na borda e a probabilidade de classificação impactam significativamente o particionamento e o tempo de inferência. Assim, mostra-se que o particionamento de DNNs com saídas antecipadas é uma solução viável para reduzir o tempo de inferência. Esse trabalho assume que a taxa de envio entre borda e a nuvem é fixa, contudo, em cenários mais realistas, a taxa é altamente dinâmica. Assim, em trabalhos futuros, pretende-se empregar técnicas para prever as condições da rede para abordar a dinamicidade da rede.

Este trabalho avaliou o impacto da distorção da imagem no tempo de inferência e no particionamento. Os resultados mostram que imagens com baixa distorção apresentam menor incerteza de classificação, facilitando sua classificação nos ramos laterais, isto é, na borda da rede. Em imagens com alto nível de distorção, utiliza-se mais a nuvem para concluir a inferência. Assim, a dissertação mostra que a distorção na imagem também afeta a decisão de particionamento.

Por fim, a dissertação avaliou, no contexto de DNNs com saídas antecipadas, como o problema de calibração afeta as decisões dos ramos laterais em classificar determinadas amostras. A dissertação mostra que aplicar um método de calibração permite refinar a capacidade de decisão dos ramos laterais, resultando em aumento de acurácia.

Referências

Bolukbasi, T., Wang, J., Dekel, O. e Saligrama, V. (2017). Adaptive neural networks for efficient inference. Em *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, p. 527–536. JMLR. org.

Guo, C., Pleiss, G., Sun, Y. e Weinberger, K. Q. (2017). On calibration of modern neural networks. Em *International Conference on Machine Learning*, p. 1321–1330. PMLR.

Hu, C., Bao, W., Wang, D. e Liu, F. (2019). Dynamic adaptive DNN surgery for inference acceleration on the edge. Em *IEEE Conference on Computer Communications (INFOCOM)*, p. 1423–1431.

Kang, Y., Hauswald, J., Gao, C., Rovinski, A., Mudge, T., Mars, J. e Tang, L. (2017). Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. Em *ACM Computer Architecture News (SIGARCH)*, volume 45, p. 615–629.

Li, E., Zeng, L., Zhou, Z. e Chen, X. (2019). Edge ai: On-demand accelerating deep neural network inference via edge computing. *IEEE Transactions on Wireless Communications*, 19(1):447–457.

Teerapittayanon, S., McDanel, B. e Kung, H.-T. (2016). Branchynet: Fast inference via early exiting from deep neural networks. Em *IEEE International Conference on Pattern Recognition (ICPR)*, p. 2464–2469.

Teerapittayanon, S., McDanel, B. e Kung, H.-T. (2017). Distributed deep neural networks over the cloud, the edge and end devices. Em *IEEE International Conference on Distributed Computing Systems (ICDCS)*, p. 328–339.