합성곱 신경망에서 동적 가지치기 모델 구현 및 적용

주조령 [1], 조인휘 [2]
[1] 한양대학교 컴퓨터소프트웨어학과 석사과정
[2] 한양대학교 컴퓨터소프트웨어학과 교수

lucinda0613@hanyang.ac.kr, iwjoe@hanyang.ac.kr

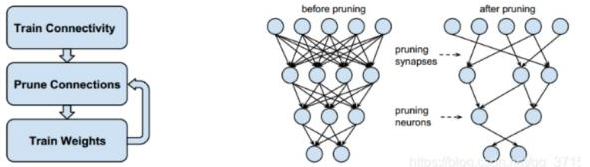
요 약

이 연구는 컴퓨팅 자원이 제한된 환경에서 딥러닝 모델의 문제를 해결하기 위해 합성곱 신경망(CNN)에서 동적 가지치기 모델의 적용을 탐구한다. 첫째, 동적 가지치기 모델의 원리와 방법에 대해 기존 방법과의 비교를 소개한다. 둘째, 기존적인 방법 동적 가지치기 모델의 구현 과정 및 결과 분석을 포함한 실험 단계를 자세히 설명한다. 실험 결과는 동적 가지치기 모델이 적절한 훈련에서 모델의 분류 성능을 효과적으로 향상시킬 수 있으며 강력한 일반화 능력을 가지고 있음을 보여준다. 마지막으로 딥러닝 방법과 기존 방법의 차이점과 장단점을 분석하고 요약하여 실제 적용에서 딥러닝 모델 배치에 유용한 탐색과 참고 자료를 제공한다. 이 연구는 딥러닝 분야에서 동적 가지치기 모델의 적용을 추가로 탐색하기 위한 중요한 이론 및 실습 기반을 제공한다.

1. 서론

딥러닝 기술의 급속한 발전으로 신경망은 컴퓨터 비전 분야에서 큰 성공을 거두었다. 그러나 모델 규모가 증가함에 따라 컴퓨팅 및 저장 요구 사항도 증가하여 임베디드 장치 및 제한된 자원 환경에서 적용이 제한된다. 이 문제를 해결하기 위해 다양한 모델 압축 및 가속 기술이 제안되었으며 그 중 동적 가지치기 모델이 많은 관심을 받았다[1].

동적 가지 치기 모델은 가지 치기 기술을 기반으로 하며 가중치 중요성 임계값을 조정하여 신경망의 가중치의 동적 조정 및 절단을 실현한다[2]. 기존적인 가지치기 방법은 훈련 후에 수행되는 반면 동적 가지치기는 훈련 중 가중치의 중요성에 따라 실시간으로 수행된다[3].



(그림 1) 동적 가지치기 모델 훈련 및 가지치기 프로세스

효율성을 향상시키기 위해 Dynamic Sparse Reparameterization[4] 동적 트리밍 알고리즘을 사용하여 가중치 중요도 임계값을 동적으로 조정하고 가중치를 재단하여 계산 부담을 줄이고 속도를 높인다. 동시에 Compressed Sparse Row(CSR)[5] 희소 매트릭스 저장 기술을 사용하여 제한된 자원 환경에 적합한 모델 저장 공간을 효과적으로 압축한다.

이 실험의 주요 목표는 컨볼루션 신경망(CNN)에서 동적 가지치기 모델의 타당성과 효율성을 검증하는 것이다[6]. 가중치 중요도 임계값을 조정하여 네트워크 가중치를 동적으로 트리밍하여 계산 부담을 줄이고 효율성을 향상시킨다. CSR 희소 매트릭스 저장 기술을 사용하면 저장 공간이 크게 줄어들고 자원이 제한된 환경에 적합한다.

다음으로 동적 가지치기 모델의 원리, 방법 및 CNN에서의 응용을 자세히 소개하고 실험적 검증 효과를 설계한다. 그 장점과 한계를 깊이 이해하고 모델 압축 및 가속 분야에 대한 참고 자료를 제공하는 것을 목표로 한다. 이 실험을 통해 CNN에서 동적 가지치기 모델의 타당성과 효율성을 입증하고 효율적인 신경망 개발을 위한 기반을 제공하며 임베디드 장치에서 딥러닝의 적용을 촉진할 것으로 기대한다. 동시에 연구자에게 모델 압축 및 가속에 대한 귀중한 경험과 지침을 제공한다.

2. 동적 가지치기 모델

현재 딥러닝 분야에서 동적 가지치기 모델은 많은 관심을 받고 있는 최첨단 방법이다. 이 모델은 희소 텐서와 가지치기 기술을 교묘하게 결합하여 신경망의 효율성과 추론 속도를 크게 향상시킨다. 핵심 아이디어는 다양한 희소성 가중치를 동적으로 생성하여 모델이 성능을 유지하면서 다양한 작업 및 자원 제약에 적응할 수 있도록 하는 것이다. 이것은 실제 적용에서 딥 러닝 모델을 배포하기 위한 더 많은 선택과 최적화 공간을 제공한다. 동적 가지치기 모델에서 주요 구성 블록에는 Dynamic Linear 및 Dynamic Conv2d가 포함되며, 이는 모델의 성능과 효율성 사이에 다리를 구축한다[7]. 동적 선형 계층으로서 Dynamic Linear는 초기 희소성 값을 기반으로 초기 희소성 값을 기반으로 희소성 텐서 또는 조밀한 텐서를 가중치로 생성하고 희소성 매개변수를 조정하여 입력 데이터의 특성에 동적으로 적응한다[8][9]. 동적 2차원 컨볼루션 레이어로 DynamicConv2d는 또한 희소성 매개변수에 따라 컨볼루션 코어로 희소 텐서 또는 조밀한 텐서를 생성하여 컨볼루션 코어의 희소성을 동적

으로 조정한다[10]. 동적 가지치기 모델은 제한된 컴퓨팅 자원 환경에서 모델의 추론 속도를 크게 향상시키고 모델의 저장 요구 사항을 줄이며 다양한 작업 및 응용 프로그램에 적합하고 딥 러닝 분야에 새로운 연구 방향을 가져올 수 있다.

3. 실험 단계

3.1 기존적인 방법

첫째, 기존의 컨볼루션 신경망 모델을 사용하여 CIFAR-10 데이터 세트에서 훈련되고 평가된 코드를 사용한다. 이 모델의 목표는 이미지를 분류하고 입력 이미지에서 범주 레이블로의 매핑 관계를 학습하여 정확한 분류 예측을 달성하는 것이다. 필요한 모듈과 함수를 가져오기 위해 Tensor Flow 및 Keras 라이브러리를 사용했다. 우리는 CIFAR-10 데이터 세트에서 훈련 세트와 테스트 세트를 로드하고 데이터 전처리를 수행하여 픽셀 값을 0 에서 1 범위로 조정했다.

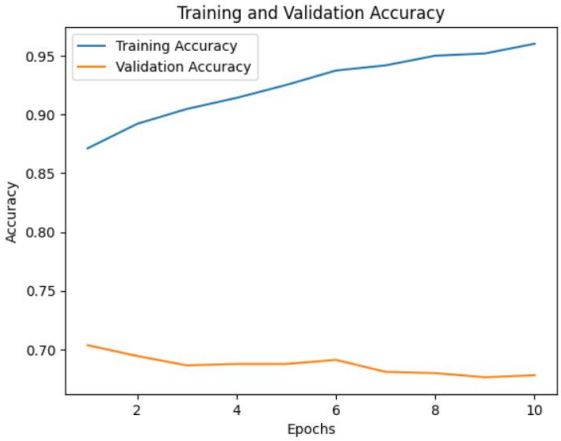
다음으로 Sequential 모델을 정의하고 일련의 컨볼루션 계층(Conv2D), 풀링 계층(MaxPooling 2D) 및 전체 연결 계층(Dense)을 추가했다. 이 모델의 아키텍처는 컨볼루션 레이어 후에 풀링 작업을 수행한 다음 출력을 평평하게 펴고 마지막으로 전체 연결 레이어에 연결한다. 마지막 레이어는 소프트맥스 활성화 함수를 사용하여 클래스의 확률 분포를 출력한다.

모델 정의가 완료된 후 ADAM 최적화 및 희소 분류 교차 엔트로피를 손실 함수로 사용하여 컴파일했다[11]. 그런 다음 모델 훈련을 위해 훈련 세트를 사용하고 훈련을 위해 10 개의 epochs 를 설정했다. 훈련이 완료된 후 테스트 세트를 사용하여 모델을 평가하고 손실 값과 정확도를 계산했다.

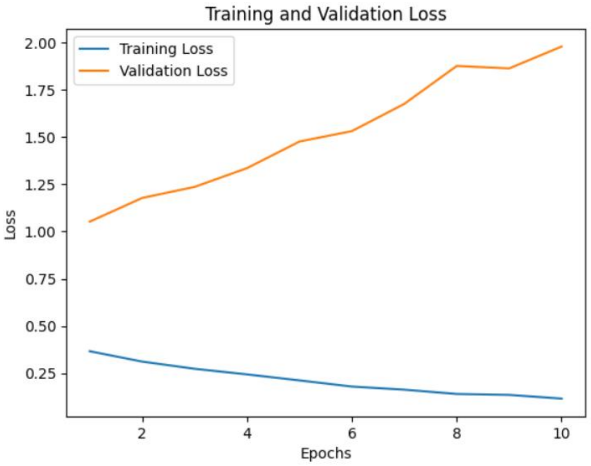
그림 2,3 에 표시된 결과 곡선은 훈련 과정 전반에 걸쳐 모델의 성능을 보여준다. 그림 2 에 '컨볼루션 신경망 모델 훈련 및 CIFAR-10 데이터 세트에 대한 훈련 및 검증 정확도 그래프' 는 모델의 정확도가 각 역량에 따라 어떻게 향상되는지 보여준다. 그것은 이미지를 올바르게 학습하고 분류하는 모델의 능력을 보여준다. 그림 3 에 '컨볼루션 신경망 모델 훈련 및 CIFAR-10 데이터 세트에 대한 훈련 및 검증 손실 그래프' 는 손실 함수의 하향 추세를 보여 모델이 매개변수를 최적화하는 정도를 나타낸다.

반복 횟수가 증가함에 따라 훈련 정확도와 검증 정확도가 증가함을 정확도 곡선에서 알 수 있다. 이것은 모델이 알려지지 않은 데이터를 잘 배우고 일반화할 수 있음을 시사한다. 그러나 특정 횟수의 epoch 후 검증 정확도가 플랫폼에 도달한 것으로 보이며, 이는 추가 훈련이 모델의 성능을 크게 향상시키지 않을 수 있음을 시사한다.

손실 곡선은 유사한 패턴을 보여주며 훈련 및 검증 손실은 시대가 진행됨에 따라 꾸준히 감소한다. 이것은 모델이 손실 함수를 최소화하기 위해 매개변수를 효과적으로 최적화했음을 시사한다. 검증 손실의 감소는 모델이 훈련 데이터에 과적합되지 않았으며 보이지 않는 예로 잘 일반화될 수 있음을 나타낸다.



(그림 2) 컨볼루션 신경망 모델 훈련 및 CIFAR-10 데이터 세트에 대한 훈련 및 검증 정확도 그래프



(그림 3) 컨볼루션 신경망 모델 훈련 및 CIFAR-10 데이터 세트에 대한 훈련 및 검증 손실 그래프

결과는 우리 모델이 10 개의 epochs 훈련 후 테스트 세트에서 약 70.42%의 정확도를 달성했음을 보여준다. 이는 모델이 테스트 세트의 이미지의 약 70.42%를 올바른 범주로 올바르게 분류할 수 있음을 의미한다. 동시에 테스트 세트에서 모델의 손실은 1.0866 으로 모델의 예측 오류를 측정하는 지표이며 일반적으로 손실 값이 낮을수록 좋다.

3.2 동적 가지치기 모델 구현

동적 가지치기 모델의 구현에서 우리는 희소 텐서와 가지치기 기술을 결합하여 신경망의 효율성과 추론 속도를 향상시키는 것을 목표로 했다. 이 접근 방식을 통해 훈련 및 추론 중에 가중치의 희소성을 조정하여 다양한 작업 및 자원 제약에 적응할 수 있다. 이것은 모델의 성능을 유지하면서 효율성을 향상시키기 위한 더 많은 유연성과 옵션을 제공한다[12].

동적 가지치기 모델의 핵심 구성 블록에는 동적 선형 층(Dynamic Linear)과 동적 컨볼루션 층(DynamicConv2d)이 포함된다. 이 층은 희소성 매개변수를 기반

으로 다양한 희소성의 가중치를 동적으로 생성한다. 초기화 중에 초기 희소성을 설정하고 희소 텐서 또는 조밀한 텐서를 가중치로 생성할 수 있다. 훈련 및 추론 과정에서 초기 희소성을 조정하여 최적의 성능과 효율성 균형을 달성하기 위해 무거운 희소성을 동적으로 제어할 수 있다.

구체적으로, 우리는 이러한 개념을 하나의 사례로 설명한다. 다층 퍼셉트론(MLP) 모델에서 동적 선형 계층을 사용하여 네트워크를 구성한다. 초기화 단계에서 동적 선형 레이어에 대한 초기 희소성을 설정하고 작업 요구 사항에 따라 가중치의 희소성을 동적으로 조정한다. 이를 통해 모델은 다양한 입력 데이터의 특성에 따라 다양한 응용 프로그램에서 더 나은 성능을 얻을 수 있다.

컨볼루션 신경망의 경우 동적 컨볼루션 층을 사용하여 모델을 구성한다. 이 레이어는 희소성 매개변수를 기반으로 다양한 희소성의 컨볼루션 코어 가중치를 생성하기 위해 유사한 방법을 사용한다. 이러한 방식으로 모델은 다양한 작업 및 자원 제한에 맞게 컨볼루션 코어의 희소성을 적응적으로 조정할 수 있다.

컨볼루션 신경망(CNN)에 동적 가지치기 모델을 사용했을 때 10 번의 훈련 주기 반복 후 일련의 흥미로운 실험 결과를 얻었다. 초기 단계에서 모델의 손실이 더 높아 테스트 세트 정확도가 50.10%에 불과했으며, 이는 모델이 데이터의 특성을 충분히 학습하지 않았음을 시사한다. 그러나 훈련이 계속됨에 따라 모델은 가중치와 매개변수를 점차 최적화하여 손실이 감소하고 테스트 세트의 정확도가 점차 향상된다[13].

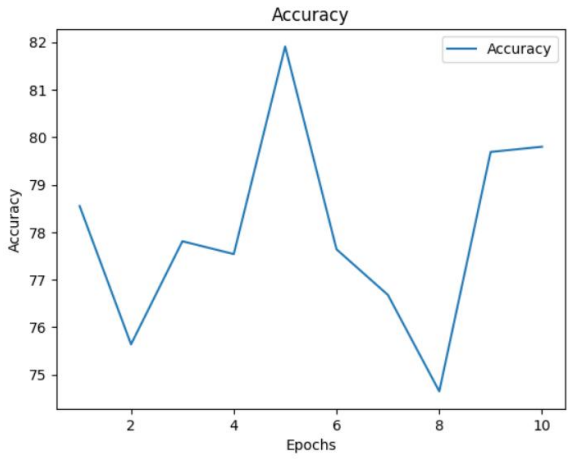
네 번째 훈련 기간 후 모델의 성능이 크게 향상되고 테스트 세트의 정확도가 73.15%에 도달하는 것을 관찰했다. 이것은 동적 가지치기 모델이 적절한 훈련에서 모델의 분류 성능을 효과적으로 향상시킬 수 있음을 보여준다. 더 고무적인 것은 훈련 기간 5 와 6 사이에서 모델이 훈련 중 가장 높은 테스트 세트 정확도를 달성하여 75.18%에 도달했다는 것이다. 이것은 가중치와 희소성을 적응적으로 조정하여 다양한 작업 및 자원 제약에 적응함으로써 동적 가지치기 모델의 능력을 더욱 강조한다.

마지막 훈련 주기 동안 모델은 77.01%의 정확도로 낮은 손실 수준을 유지했다. 이것은 훈련의 후반 단계에서도 모델이 성능을 유지할 수 있고 명백한 과적합이 없음을 보여준다. 그러나 우리는 또한 일부 훈련 주기 동안 모델의 성능이 약간 감소했는데, 이는 과적합 현상으로 인한 것일 수 있다.

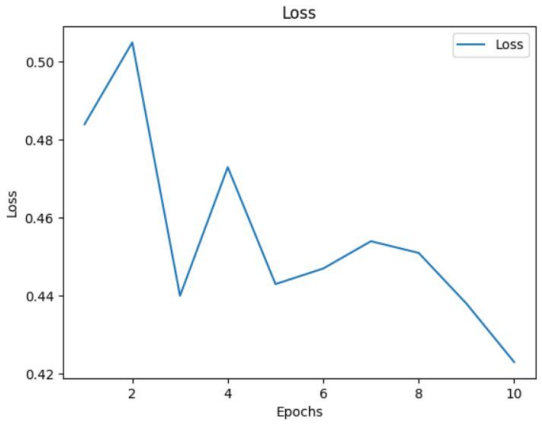
위의 코드를 기반으로 동적 가지치기 모델의 훈련 및 검증을 위한 정확도 곡선과 손실 곡선을 그렸다. 이러한 곡선은 훈련 중 모델의 성능 변화를 분석하는 데 도움이 될 수 있다.

정확도 곡선은 훈련 라운드 수에 따라 훈련 세트 및 검증 세트에 대한 모델의 정확도 변화를 보여준다. 정확도 곡선에서 훈련 세트의 정확도가 점차 증가함을 관찰할 수 있으며, 이는 모델이 훈련 과정에서 점차 더 나은 특성 표현 및 분류 능력을 습득했음을 나타낸다. 그러나 검증 세트의 정확도는 크게 향상되지 않은 것으로 보이며, 이는 모델이 어느 정도 과적합되어 보이지 않는 데이터로 일반화되지 않았음을 나타낼 수 있다.

손실 곡선은 훈련 라운드 수에 따라 훈련 세트 및 검증 세트에서 모델의 손실 값의 변화를 보여준다. 손실 곡선에서 훈련 세트의 손실 값이 점차 감소하는 것을 관찰할 수 있으며, 이는 모델이 학습 과정에서 점차적으로 오류를 줄였음을 나타낸다. 그러나 검증 세트에 대한 손실 값은 뚜렷한 하향 추세를 보이지 않았으며, 이는 또한 위에서 언급한 과적합 가능성을 뒷받침한다.



(그림 4) 동적 가지치기 모델에 대한 훈련 및 검증 정확도 곡선



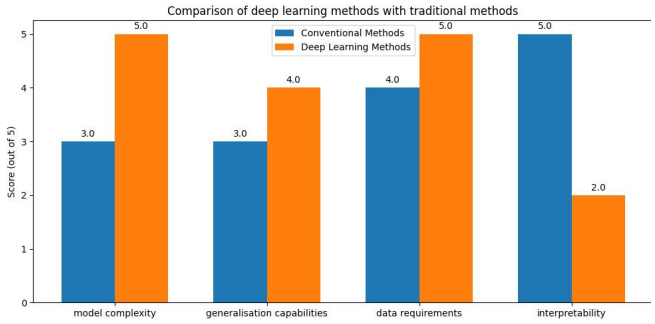
(그림 5) 동적 가지치기 모델에 대한 훈련 및 검증 손실 곡선

요약하면, 이 실험을 통해 컨볼루션 신경망의 성능을 향상시키는 동적 가지치기 모델의 잠재력을 명확히 했다. 과적합을 피하기 위해 훈련 전략을 추가로 최적화하는 등 여전히 개선의 여지가 있지만, 이 실험은 실제 응용 분야에서 딥러닝 모델 배포에 유용한 탐색과 참조를 제공한다. 앞으로 우리는 더 나은 성능과 효율성 균형을 달성하기 위해 동적 가지치기 모델의 모든 측면에 대한 심층 연구를 계속할 것이다.

4. 분석 및 요약

이 프로젝트에서 우리는 CIFAR-10 데이터 세트를 사용하여 Wide ResNet 모델을 훈련하고 성능을 기존 방법과 비교하고 모델 복잡성, 일반화 능력, 데이터 요구 사항 및 해석 가능성의 4 가지 측면에서 각각 점수를 매겼다. 채점 결과는 해당 텍스트 해석과 함께 히스토그램으로 표시된다.

모델 복잡성 측면에서 우리는 전통적인 방법과 딥러닝 방법을 별도로 평가했다. 그림과 같이 딥러닝 방법은 모델 복잡성 점수가 높아 모델 구조와 매개변수가 더 복잡함을 나타낸다. 대조적으로, 전통적인 방법은 모델 복잡성이 낮고 주로 수동으로 설계된 기능과 알고리즘 선택에 의해 제한된다.[14]



(그림 6)딥러닝 방법과 기존 방법의 평가 대비

일반화 능력은 모델이 본 적이 없는 데이터에서 성능을 측정하는 능력이다. 기존적인 방법과 딥 러닝 방법에 대한 평가를 통해 딥 러닝 방법이 일반화 능력에서 더 높은 점수를 받았다는 것을 발견했으며, 이는 다양한 데이터 패턴 및 변동에 더 잘 적응할 수 있음을 나타낸다. 그러나 기존적인 방법은 수동으로 설계된 기능이 데이터의 잠재적 패턴을 완전히 포착하지 못하여 성능이 좋지 않을 수 있다.

데이터 요구 사항은 모델 교육에 필요한 마커 데이터의 양을 나타낸다. 딥 러닝 방법은 일반적으로 대규모 데이터에서 이점을 얻을 수 있지만 더 많은 마커 데이터에 대한 요구 사항이 더 높다. 대조적으로, 기존적인 방법은 일반적으로 소규모 데이터에서도 잘 수행되지만 기능을 추출하려면 더 많은 전문가 지식이 필요할 수 있다.

해석 가능성은 모델 결정 프로세스가 이해하기 쉽고 해석하기 쉬운지 여부를 평가하는 지표이다. 기존적인 방법은 기능 선택 및 알고리즘 설계를 수동으로 수행하기 때문에 종종 해석 가능한 결과를 제공한다. 대조적으로, 딥러닝 방법은 설명성이 약하고 특히 복잡한 모델은 의사 결정 과정을 설명하기 어려운 블랙박스에 가깝다.

위의 평가를 통해 우리는 다양한 측면에서 기존적인 방법과 딥러닝 방법의 장단점을 명확하게 볼 수 있다. 특정 작업 및 요구 사항에 적합한 방법을 선택하는 것은 실제 상황에 따라 종합적으로 고려되어야 한다.

5. 결론

이 연구는 동적 가지치기 모델이 컨볼루션 신경망의 효율성과 추론 속도를 향상시키는데 상당한 이점이 있음을 발견했다. 실험 결과는 동적 가지치기 모델이 모델의 분류 성능을 향상시키는데 효과적임을 보여주었다. 과적합을 피하기 위해 훈련 전략을 최적화할 필요가 있지만, 동적 가지치기 모델은 신경망의 효율성과 추론 속도 향상에 새로운 관점을 제공한다. 이 연구는 실제 응용 분야에서 유용한 탐색과 참조를 제공하며, 더 나은 성능과 효율성 균형을 위해 동적 가지치기 모델에 대한 심층 연구를 계속할 것이다.

참고문헌

[1] 高晗, 田育龙, 许封元, & 仲盛. (2020). Survey of deep learning model compression and acceleration. Journal of Software, 32(1), 68-92.

[2] Xu, P., Cao, J., Shang, F., Sun, W., & Li, P. (2020). Layer pruning via fusible residual convolutional block for deep neural networks. *arXiv preprint arXiv:2011.14356*.

[3] Han, Y., Huang, G., Song, S., Yang, L., Wang, H., & Wang, Y. (2021). Dynamic Neural Networks: A Survey. *arXiv preprint arXiv:2102.04906*.

[4] Mostafa, H., & Wang, X. (2019). Parameter Efficient Training of Deep Convolutional Neural Networks by Dynamic Sparse Reparameterization. *arXiv preprint arXiv:1902.05967*.

[5] Liu, X., Pool, J., Han, S., & Dally, W. J. (2018). Efficient sparse-winograd convolutional neural networks. *arXiv preprint arXiv:1802.06367*.

[6] Guo, Y., Yao, A., & Chen, Y. (2016). Dynamic Network Surgery for Efficient DNNs. *arXiv preprint arXiv:1608.04493*.

[7] Liu, J., Xu, Z., Shi, R., Cheung, R. C., & So, H. K. (2020). Dynamic sparse training: Find efficient sparse network from scratch with trainable masked layers. *arXiv preprint arXiv:2005.06870*.

[8] Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N., & Peste, A. (2021). Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks. *arXiv preprint arXiv:2102.00554*.

[9] Atashgahi, Z., Pechenizkiy, M., Veldhuis, R., & Mocanu, D. C. (2023). Adaptive Sparsity Level during Training for Efficient Time Series Forecasting with Transformers. *arXiv preprint arXiv:2305.18382*.

[10] Sokar, G., Mocanu, E., Mocanu, D. C., Pechenizkiy, M., & Stone, P. (2021). Dynamic sparse training for deep reinforcement learning. *arXiv preprint arXiv:2106.04217*.

[11] Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv preprint arXiv:1502.03167*.

[12] Huang, Q., Zhou, K., You, S., & Neumann, U. (2018). Learning to Prune Filters in Convolutional Neural Networks. *arXiv preprint arXiv:1801.07365*.

[13] Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

[14] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.