특집원고

비 중심화된 데이터에 따른 비 동기화 연합학습 알고리즘

포항공과대학교 ▌박찬호·이승훈·이남윤

1. 서 론

연합학습(Federated learning)은 서버와 여러 엣지 디바이스(Edge device)들 사이에서 분산되어있는 정보를 직접 공유하지 않으면서 전체 모델을 학습할 수 있는 기계학습(Machine learning)의 일종이다. 자신만의 통신 장치이자 연산 장치인 PC 혹은 스마트폰을 통해 많은 작업이 처리되는 현대 사회에서, 연합학습은 개개인의 전자기기를 활용할 수 있다는 점에서 굉장히 효율적인 방법이다 [1]. 이미 많은 분야에서 연합학습의 장점들을 인정하고 이용하기 시작했다. 대표적인 예시로 Google에서는 연합학습을 이용하여 상대적으로 스마트폰 사용량이 적은 새벽 시간에 정보를 수집해서 Gboard 프로그램을 개발하였고 [2], 보안이 중요한 의학 분야에서도 연합학습을 이용한 의료 기기 개발을 위해 노력 중이다 [3]. 더 발전된 기계학습 기술을 위하여 연합학습에 대한 높은 이해도를 가지는 것은 불가피하다.

연합학습의 알고리즘을 간단히 소개하자면, 중앙 서버와 엣지 디바이스에서의 알고리즘 총 2단계로 나눌 수 있다. 먼저, 연합학습에 참여하는 엣지 디바이스들이 중앙 서버에서 가지고 있는 초깃값의 학습 매개변수를 무선 네트워크를 이용하여 다운로드한다. 각 엣지 디바이스에서는 편향된 데이터를 학습하고, 학습에 사용한 손실함수(Loss function)를 이용하여 새로운 학습 매개변수 혹은 기울기를 계산해 중앙 서버에 업로드한다. 중앙 서버에서는 일반화된 학습 모델을 얻기 위해 각 엣지 디바이스로부터 온 정보를 집계(Aggregation)한다. 이후 최적화 기법(Optimization method)을 이용해 새로운 학습 매개변수를 얻고, 학습 모델을 업데이트한다. 위의 알고리즘을 학습 모델이 수렴할 때까지 반복하여 최종 학습 모델을 얻는다 [4].

† 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 ICT혁신인재4.0사업의 연구결과로 수행되었음 (IITP-2020-2020-0-01822)

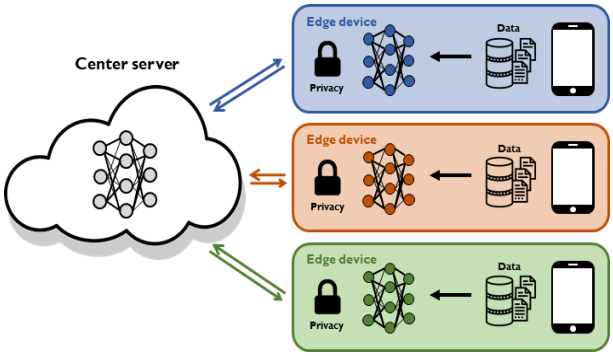

그림 1 연합학습 알고리즘

연합학습이 학계에서 주목받는 데에는 3가지의 큰 장점이 있기 때문이다. 첫 번째로 사적인 정보를 공유하지 않는다는 점이다. 연합학습이 나오기 전에 존재했던 기계학습 방법인 분산학습(Distributed learning)은 엣지 디바이스들이 모두 중심화된 데이터를 가져야 했다. 연합학습을 이용하면 엣지 디바이스에서 사적인 정보를 가지더라도 학습이 가능하다. 또한 중앙 서버에 데이터를 직접 공유하지 않고 학습 매개변수를 계산하여 서로 주고받기에 정보에 대한 보안성도 지킬 수 있다. 두 번째 장점은 중앙 서버의 파워 비중이 줄어든다는 점이다. 하나의 연산 장치에 다량의 데이터를 저장하여 학습했던 초창기 기계학습 방법과 비교해, 연합학습은 다량의 데이터가 여러 엣지 디바이스들로 분산되어 중앙 서버의 연산량이 현저히 줄어든다. 연산량을 여러 엣지 디바이스들로 나눌 수 있어 학습에 걸리는 시간 역시 효율적으로 줄일 수 있다. 마지막으로 사용자가 많아질수록 다양한 방향성의 데이터들이 모이기에 편향된 데이터들로 학습을 하더라도 서버에서 얻는 학습 모델은 한쪽으로 편향되지 않은 결과를 얻을 수 있다 [5].

연합학습을 더 일반화하거나 장점들을 더 강화하기 위해 연구들이 활발히 진행 중이다. 연합학습을 통해서 생기는 이전의 기계학습 방식과는 다른 문제점들도 있기에, 이를 해결 또는 완화하기 위해 연합학습의

알고리즘에 대한 다양한 접근이 이루어지고 있다. 본 원고에서는 연합학습의 중앙 서버와 엣지 디바이스들 사이에서 정보 전달이 지연되는 생기는 문제점들에 대해 논하고, 이에 대한 여러 아이디어를 소개한다.

2. 동기화/비 동기화 연합학습의 등장 배경

연합학습은 데이터를 여러 엣지 디바이스에 나누어 학습하기에 각자의 데이터를 함축한 정보들을 잘 집계하여 새로운 학습 모델을 만들어줄 중앙 서버가 필요하다. 이 과정에서 중앙 서버와 엣지 디바이스 간의 학습 매개변수 혹은 기울기(Gradient)을 무선 네트워크를 이용하여 공유해야 한다. 엣지 디바이스에서 중앙 서버에 보내는 정보들이 정해진 시간 내에 모두 도착하면 빠른 속도로 연합학습 알고리즘이 진행되겠지만, 실제로는 그렇지 않다. 그 이유로 먼저 엣지 디바이스들은 높은 샘플링 주파수를 갖기 때문에 엣지 디바이스에 따라 데이터의 양이 크게 차이 나는 경우가 생기게 된다. 학습량이 많아질수록 학습 시간은 당연히 증가하게 되고, 따라서 엣지 디바이스에 따라 학습 시간에 큰 차이가 생긴다. 둘째로, 앞서 설명했던 것처럼 중앙 서버와 엣지 디바이스들이 온라인으로 정보 공유를 하기에, 상황에 따라 네트워크의 상태가 좋지 않은 링크가 존재한다면 장치 간의 정보 전달 속도에 큰 차이가 생긴다. 간단한 예시로 연합학습에 참여할 엣지 디바이스들 중 상당수가 축구 경기장에 있는 사용자들의 스마트폰이라고 하면, 경기장에 있는 사용자와 아닌 사용자들 간의 환경에 큰 차이가 존재한다. 네트워크의 품질이 상황에 따라 달라질 수 있기에 그에 따라 정보 전달이 지연되는 문제점 역시 생기게 된다. 데이터 역시 엣지 디바이스에 따라 방향성이 달라서 그에 따른 학습 속도가 많이 다르게 되고, 엣지 디바이스의 연산 능력이나 배터리에 따라서도 학습 속도가 달라진다. 따라서 각 엣지 디바이스로부터 보낸 정보가 중앙 서버에 도달하는 시간이 달라지는 요인이 다양하며, 연합학습을 실생활에 적용하기 위해서는 반드시 고려해야 하는 문제점이다 [6].

각 엣지 디바이스에서 중앙 서버에 정보 전달하는 시간이 달라지면 학습 효율에 문제가 생기게 된다. 모든 엣지 디바이스에서 정해진 시간 내에 정보를 전달하게 되면, 중앙 서버에서 다양한 방향성의 데이터들을 잘 합산하여 새로운 학습 모델을 업데이트할 수 있다. 하지만 정해진 시간 내에 정보 전달이 되지 않은 엣지 디바이스들이 생기면 그 장치들에서 오는 정보를 기다리거나, 그 장치들에 대한 정보 없이 중앙 서버에서 집계하는 두 가지의 방법을 고려할 수 있다. 도착이 지연되는 정보들을 기다리게 되면 중앙 서버와 엣지 디바이스 간의 연합학습 알고리즘이 한 번 시행되는 데 걸리는 시간이 매우 증가한다. 만약 지연되는 정보들을 기다리지 않고 중앙 서버에서 도달한 정보들을 집계하면, 중앙 서버에 도착하지 않은 엣지 디바이스들의 데이터에 대해서는 학습되지 않은 채로 새로운 학습 모델을 업데이트한다. 이 경우는 각 엣지 디바이스가 편향된 데이터를 가진 상황에서 모든 장치에 공평하게 학습된 학습 모델을 만들 수 없다. 따라서 이 두 방법에 대해 학습 모델의 정확성과 학습하는 데 걸리는 시간 사이의 트레이드-오프 관계가 생긴다. 이렇게 모든 엣지 디바이스들로부터 중앙 서버에 오는 정보들을 기다리는 알고리즘을 동기화 연합학습(Synchronous Federated Learning)이라고 하고, 정해진 시간 내에 도달하지 않은 정보를 제외하고 중앙 서버에서 학습 모델을 업데이트하는 알고리즘을 비 동기화 연합학습(Asynchronous Federated Learning)이라고 한다 [7].

동기화 연합학습의 문제점은 결국 학습 시간이 길어진다는 점인데 [8], 이 문제를 해결하기 위해서는 엣지 디바이스들의 네트워크의 성능을 향상하거나,

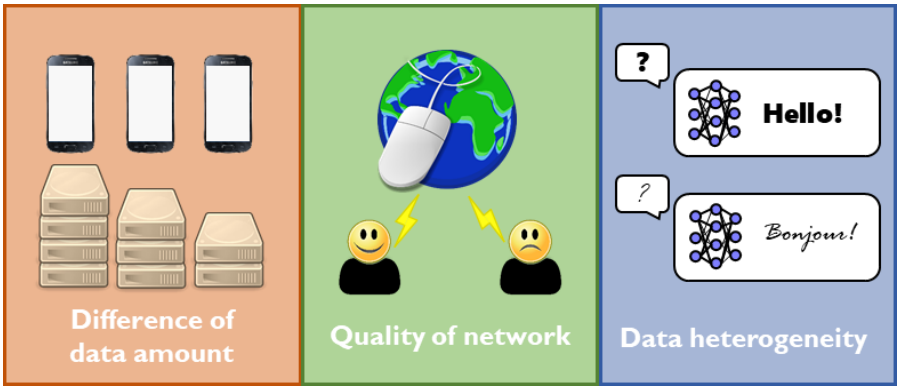

그림 2 연합학습 내의 정보 전달 시간 차이를 만드는 요인

엣지 디바이스에서 데이터에 대한 연산량이 장치들끼리 최대한 비슷하도록 맞춰주는 방식을 고려할 수 있다. 하지만 이 작업을 통해 모든 장치가 각자만의 데이터를 가진다는 특색이 옅어지고, 동기화 연합학습의 성능을 업그레이드하기 위해서 알고리즘을 보완하기보다는 무선 통신에서의 기술적인 변화가 필요하다. 이에 반해 비 동기화 연합학습에 관해 최근 진행되는 연구에서는 연합학습 알고리즘에서의 변화만으로도 성능을 개선하는 아이디어들이 나오고 있다. 본 원고에서는 비 동기화 연합학습의 성능 개선을 위한 새로운 알고리즘들에 대해 살펴본다.

3. 비 동기화 연합학습 알고리즘

비 동기화 연합학습 알고리즘은 정해진 시간 동안에만 엣지 디바이스들로부터 정보를 받기 때문에 모든 정보가 중앙 서버에 도착할 때까지 기다리는 동기화 연합학습보다 월등히 빠른 학습 속도를 얻을 수 있다. 그러나 학습하는 동안 특정 엣지 디바이스들에게서 계속해서 정보를 받지 못한다면 그 장치들의 데이터는 학습이 되지 않고, 학습 모델의 일반성을 잃는다. 정해진 시간 내에 도착하지 않은 장치의 정보들을 특정 값으로 대체하여 학습하거나 중앙 서버의 집계 알고리즘에 변화를 주는 등 학습 모델의 정확도를 높이기 위해 연구가 활발히 진행되고 있다. 아래부터는 연구되고 있는 비 동기화 연합학습 알고리즘들에 관해 간단히 소개한다.

먼저 소개할 알고리즘은, 중앙 서버에서 업데이트한 학습 매개변수를 다운로드할 수 있는 엣지 디바이스들만 우선 받아서 각자의 데이터로 학습한다. 학습하여 얻은 새로운 학습 매개변수나 관련 값을 중앙 서버에 업로드하는데, 중앙 서버에서는 이 중 가장 빨리 도달한 엣지 디바이스의 정보만 선택하여 새로운 학습 매개변수로 업데이트한다. 따라서 이 연합학습 알고리즘을 한 번 시행하는 데 걸리는 시간이 기존 동기화 연합학습의 알고리즘에 비해 월등하게 줄어든다. 다만 알고리즘이 한 번 시행될 때 하나의 엣지 디바이스에 대해서만 학습 모델이 업데이트되기 때문에 학습 모델의 정확도를 높이기 위해서는 다른 알고리즘에 비해 여러 번의 학습 모델의 업데이트가 필요하다. 더욱이 한 번의 알고리즘 시행에 하나의 엣지 디바이스만 학습되기에, 학습이 진행되는 시간 내내 특정한 엣지 디바이스의 품질이 나쁘다면 최종적으로 얻을 수 있는 학습 모델이 모든 엣지 디바이스들에 의해 업데이트되는 것이 아니라서 일반성을 잃는다. 이 알고리즘의 연구자들은 정확도를 높이기 위하여 중앙 서버에서 학습 모델을 업데이트할 때 기능학습(Feature learning)을 이용하여 각 데이터의 특징을 잘 구분하여 학습하도록 설계하였다. 더불어 각 엣지 디바이스의 데이터 연관성을 계산하는 정칙화(Regularization) 항을 추가한 손실함수를 연산에 이용하여 학습하여 학습 모델의 정확도를 높였다. 해당 알고리즘의 연구자는 3가지 종류의 데이터를 이용하여 기존 연합학습 알고리즘과 비교한 결과, 해당 알고리즘이 학습 속도가 FedAvg을 이용한 동기화 연합학습 알고리즘과 비교해 월등히 빠르고, 학습 모델의 정확도 역시 좋은 결과를 얻을 수 있음을 확인하였다 [6].

비 동기화 연합학습에 대한 또 다른 획기적인 알고

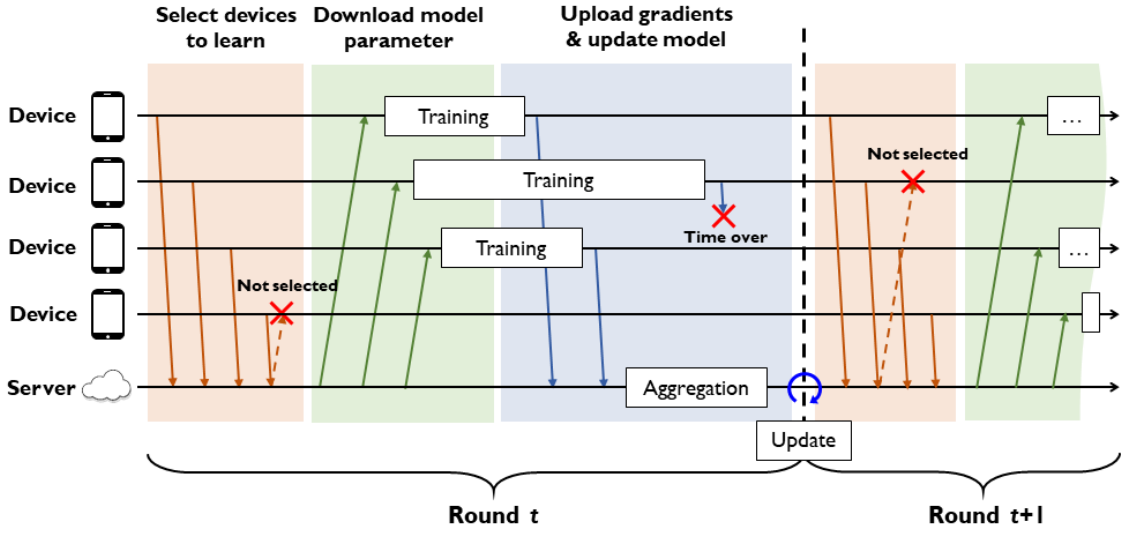

그림 3 비 동기화 연합학습 알고리즘 예시

리즘은, 각 엣지 디바이스에 존재하는 데이터에 중앙 서버에는 공유되지 않는 무작위 난수 생성 행렬(Random generator matrix)과 가중치 행렬(Weight matrix)을 곱하여 연합학습 알고리즘이 시행되기 전에 중앙 서버에 전달한다. 중앙 서버에서는 부호화된 데이터를 받기 때문에 각 엣지 디바이스들이 가지는 데이터에 대한 보안성은 지켜진다. 해당 연구에서는 각 엣지 디바이스들이 가지는 데이터의 양에 따라서 손실함수에 대한 기울기를 연산하는 시간과 기울기를 중앙 서버에 업로드하는 시간의 최적값을 계산한다. 정보가 중앙 서버에 도달하는 시간은 기하 분포(Geometric distribution)를 따르는 확률 변수(Random variable)로 설계하였다. 가중치 행렬에 이용되는 행렬 성분 값은 각 엣지 디바이스가 최적 시간 내에 정보가 전달되지 않을 확률로 설계하였다. 해당 알고리즘은 비 동기화 연합학습 알고리즘과 마찬가지로 위에서 정한 최적 시간 내에 각 엣지 디바이스에서 보낸 기울기 정보가 도달하지 않으면 그 정보는 이용하지 않고 중앙 서버에서 집계한다. 추가로 진행되는 과정은 제일 처음에 보냈던 부호화된 데이터를 이용하여 그 데이터에 따른 기울기를 계산하고, 각 엣지 디바이스에서 보낸 기울기를 집계하여 구한 값과 더한다. 이 과정을 거치게 되면, 최종적으로 계산한 기울기 값의 평균이 모든 엣지 디바이스에서 각자의 데이터를 이용하여 연산한 기울기들의 총합과 같아진다 즉, 기울기를 평균적인 관점에서 보았을 때 해당 알고리즘과 동기화 알고리즘을 이용한 학습 모델의 정확도가 같아진다. 다른 비 동기화 연합학습 알고리즘과 비교했을 때, 해당 알고리즘은 초기에 각 엣지 디바이스의 데이터가 코딩되어 중앙 서버에 업로드되기 때문에 단순한 비 동기화 연합학습 알고리즘보다 공유되는 정보량이 더 많아진다. 더불어 데이터에 코딩되는 행렬을 중앙 서버에서 알게 되면 결국 모든 장치들의 데이터를 중앙 서버에서 알게 된다. 다만 코딩되는 정보에 대한 보안이 잘 이루어진다면 기존 동기화 연합학습 알고리즘에 비해 월등한 학습 속도를 얻을 수 있고, 정확도 또한 비슷하게 얻을 수 있다. 코딩을 통해 엣지 디바이스들의 보안을 지키면서도 학습 속도와 학습의 정확도를 높일 수 있는 점에서 큰 장점이 있는 연구이며, 코딩 방법에 관한 연구가 계속해서 진행 중이다 [9].

마지막으로 소개할 비 동기화 연합학습 알고리즘은, 한 엣지 디바이스로부터 정보가 도달하면 그 정보에 대해 중앙 서버에서 바로 새로운 학습 모델을 업데이트하고 다시 그 엣지 디바이스에게 업데이트된 학습 모델을 공유한다. 따라서 모든 엣지 디바이스들이 중앙 서버와 정보를 공유하면 그 정보를 이용해 즉시 학습 모델을 유동적으로 업데이트한다. 이전 방법들과의 차이는 엣지 디바이스에서 중앙 서버에 보내는 정보가 학습 매개변수라는 점이고, 중앙 서버에서는 받은 학습 매개변수를 이전 시간의 학습 매개변수와 선형 결합(Linear combination)을 통해 업데이트한다. 주의해야 할 점은, 모든 엣지 디바이스로부터 받는 학습 매개변수를 이용하여 바로 업데이트하기 때문에 중앙 서버에서 업데이트가 이미 진행 중이어서 그 시간 동안 업데이트를 기다리게 되는 엣지 디바이스가 발생한다. 그 시간에 대한 보상을 선형 결합을 할 때 사용되는 계수를 기다린 시간에 따라 달라지게 하여 해결한다. 모든 엣지 디바이스들에 대해 유동적으로 중앙 서버에서 학습 모델을 업데이트하기에 학습 모델에 대한 정확성이 어느 정도 보장되고, 학습 시간 역시 동기화 연합학습의 알고리즘에 비해 적어진다. 학습 매개변수의 수가 많아질수록 중앙 서버에서 연산하는 양이 많아지고 연산 시간이 길어지기 때문에, 업데이트를 기다리는 엣지 디바이스들이 많아짐에 따라 이에 대한 알고리즘의 복잡도가 높아질 수 있는 단점이 있다. 데이터에 대한 학습 모델이 복잡해질수록 스케줄링이 어려워진다는 문제를 해결해야 한다 [10].

앞서 설명한 알고리즘 이외에도 비 동기화 연합학습의 학습 시간이나 학습 모델의 정확도를 높이기 위한 다양한 방법이 존재한다. 더 나아가야 할 점은, 연구의 편의성을 위해 손실함수를 볼록 함수로 설계한 내용이 많지만, 데이터에 따라 적합한 손실함수가 달라질 수 있다. 데이터에 부합하는 함수 혹은 최적화 기법을 사용하지 않으면 학습의 효율이 떨어질 수 있으므로, 더 일반화된 연합학습 알고리즘이 필요하다.

4. 결 론

연합학습은 엣지 디바이스들이 서로 다른 방향성의 데이터를 가지게 되면서 학습 시간과 업로드에 걸리는 시간이 달라진다는 문제점이 있다. 이를 해결할 방법으로 동기화 연합학습과 비 동기화 연합학습 두 종류의 알고리즘이 고안되었다. 최근에 진행되는 여러 연구를 통해 비 동기화 연합학습 알고리즘이 빠른 학습 속도를 가지며 학습 모델의 정확도 또한 동기화 연합학습과 비슷한 수준을 얻을 수 있는 것을 확인하였다. 나아가 다양한 데이터에 대해 연합학습을 적용하기 위해 더 일반화된 연합학습 알고리즘에 관해 연구해야 한다.

참고문헌

[1] Imteaj, A., & Amini, M. H. (2019, December). Distributed sensing using smart end-user devices: pathway to federated learning for autonomous IoT. In 2019 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 1156-1161). IEEE.

[2] Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., ... & Ramage, D. (2018). Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604.

[3] Brisimi, T. S., Chen, R., Mela, T., Olshevsky, A., Paschalidis, I. C., & Shi, W. (2018). Federated learning of predictive models from federated electronic health records. International journal of medical informatics, 112, 59-67.

[4] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics (pp. 1273-1282). PMLR.

[5] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020, June). How to backdoor federated learning. In International Conference on Artificial Intelligence and Statistics (pp. 2938-2948). PMLR.

[6] Chen, Y., Ning, Y., Slawski, M., & Rangwala, H. (2019). Asynchronous Online Federated Learning for Edge Devices with Non-IID Data. arXiv preprint arXiv:1911.02134.

[7] Sprague, M. R., Jalalirad, A., Scavuzzo, M., Capota, C., Neun, M., Do, L., & Kopp, M. (2018, September). Asynchronous federated learning for geospatial applications. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 21-28). Springer, Cham.

[8] Chai, Z., Fayyaz, H., Fayyaz, Z., Anwar, A., Zhou, Y., Baracaldo, N., & Cheng, Y. (2019). Towards taming the resource and data heterogeneity in federated learning. In 2019 {USENIX} Conference on Operational Machine Learning (OpML 19) (pp. 19-21).

[9] Dhakal, S., Prakash, S., Yona, Y., Talwar, S., & Himayat, N. (2019, December). Coded federated learning. In 2019 IEEE Globecom Workshops (GC Wkshps) (pp. 1-6). IEEE.

[10] Xie, C., Koyejo, S., & Gupta, I. (2019). Asynchronous federated optimization. arXiv preprint arXiv:1903.03934.

약 력

박 찬 호

2020 포항공과대학교(POSTECH) 전자전기공학과 공학사
2020~현재 포항공과대학교(POSTECH) 전자전기공학과 석사과정
관심분야: MIMO 시스템, 머신러닝

이 승 훈

2018 포항공과대학교(POSTECH) 전자전기공학과 공학사
2020 포항공과대학교(POSTECH) 전자전기공학과 공학석사
2020~현재 포항공과대학교(POSTECH) 전자전기공학과 박사과정
관심분야: MIMO 시스템, 신호처리, 머신러닝

이 남 윤

2006 고려대학교 전파통신 공학과 공학사
2008 한국과학기술원 전자공학과 공학석사
2014 텍사스주립대학 (오스틴) 전자컴퓨터공학과 공학박사
2008~2011 삼성종합기술연구원 선임연구원
2014~2015 노키아 연구센터 (미국 캘리포니아 버클리) 책임연구원
2015~2016 인텔 연구소 (미국 캘리포니아) 책임연구원
2016~현재 포항공과대학교 전자전기공학과 부교수
2019~현재 편집위원 (IEEE Transactions on Communications, IEEE Transactions on Wireless Communications, IEEE Communications Letters)
관신 분야: 통신 이론, 머신 러닝