**모델 선정**

라즈베리파이 딥러닝 모델 선정

2014312971

이영건

**목표**

* 라즈베리파이에서 활용할 수 있는 모델을 찾고 비교하여 추후 학습 시 참고한다.

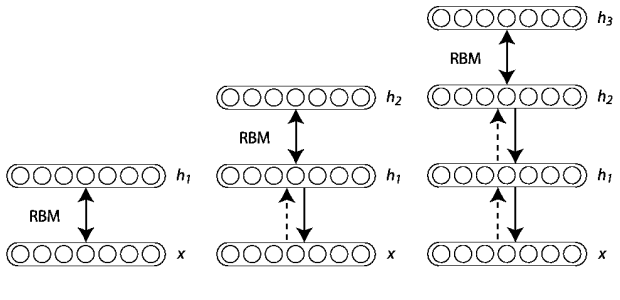
\*Quantize(양자화)

라즈베리파이 같이 작은 메모리와 파워, GPU를 가진 소형 기기에서 딥러닝을 하기위해 32bit float 형태의 노드와 가중치를 8bit 정수로 단순화하여 진행하는 것 -> 각 파일의 사이즈를 75%줄일 수 있다.

\*라즈베리파이의 processor는 느리고, RAM도 작기 때문에 노트북이나 기타 장비에서 모델을 학습시키고 라즈베리파이에서는 학습된 모델을 이용한 데이터 판별을 하는 것을 목표로 한다.

**모델 분류와 정리**

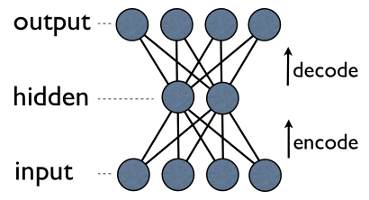
**. Deep Belief Network**



RBM(Restricted Boltzmann Machine)은 그래프로 표현된 확률모델입니다. 신경망과 비슷한 구조인데 이 RBM를 여러층을 쌓고 비지도학습으로 Pre-training을 합니다.

이렇게 하면 비슷한 특징들로 묶여진 신경망의 초기 가중치를 정할 수 있고 그 다음에는 완전연결 신경망을 사용하여 지도학습을 수행합니다.

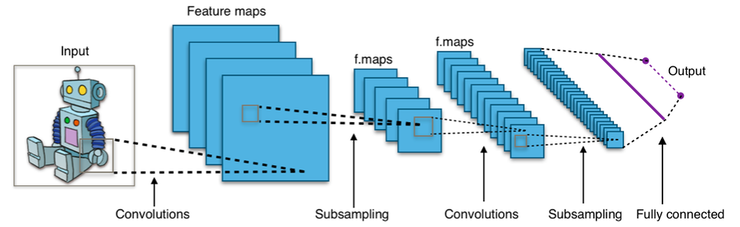
**2. Autoencoder**



오토인코더는 하나의 히든레이어가 있는 신경망인데 입력값이 그대로 출력값이 됩니다. 대신 히든레이어가 입력값을 압축하여 특징을 추출할 수 있습니다.

히든레이어을 다시 입력값으로 연결하여 여러층의 오토인코더를 쌓은 다음  DBN와 마찬가지로 비지도학습으로 Pre-training을 합니다.

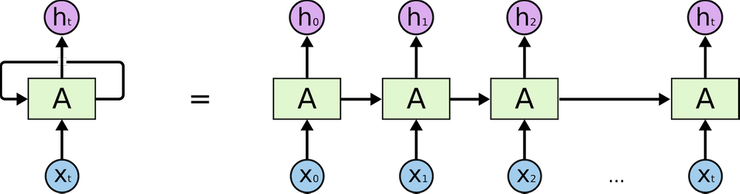
**3. Convolutional Neural Network**



가장 많이 알려진 딥러닝 방법으로 특히 이미지 처리에 적합합니다. 입력을 필터로 합성곱하여 특징을 추출하고 이런 합성곱 레이어를 여러 계층으로 연결합니다.

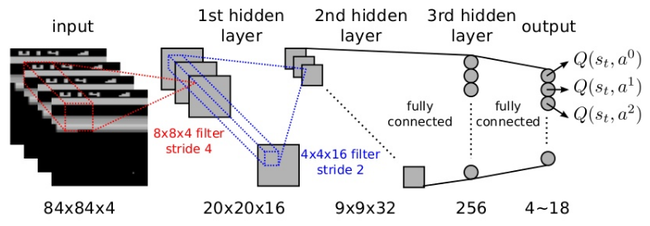
합성곱 레이어를 지날때마다 저수준의 특징들이 점차 고수준의 특징들로 만들어집니다. 그리고 마지막에는 완전연결 레이어로 최종 결과를 학습합니다.

**4. Recurrent Neural Network**



RNN의 가장 큰 특징은 시간에 따른 순서를 기억한다는 것입니다. 신경망의 출력이 다른 신경망과 연결되어 있고 이렇게 여러개의 신경망으로 이어져 있습니다. 단어의 순서에 따른 연결이 고려되는 자연어처리에서 많이 사용되고 있습니다.

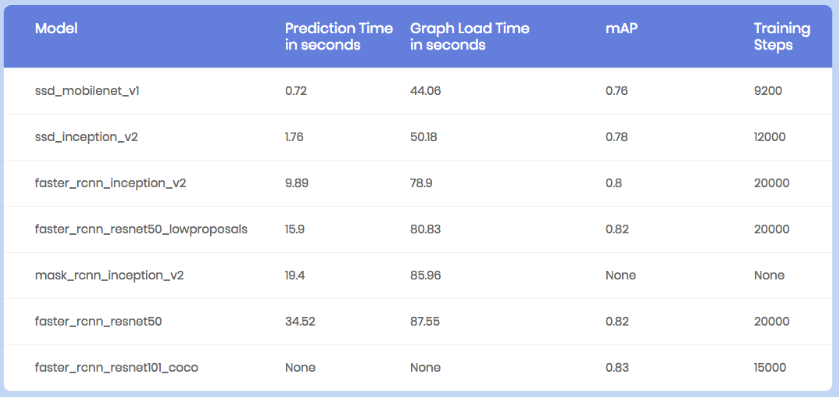
**5. Deep Q-Network**



강화학습은 현재 상태 S에서 행동 a를 결정하고 그에 따른 보상을 받아 행동을 수정합니다. Q테이블은 각 상태집합에서 행동에 따른 우선순위가 있는 테이블입니다.

Q(S, a)는 상태 S에서 a라는 행동을 했을때의 이득값입니다. 현재 상태의 모든 행동 a에 대해서 Q값을 구해 가장 높은 우선순위의 행동을 수행합니다.

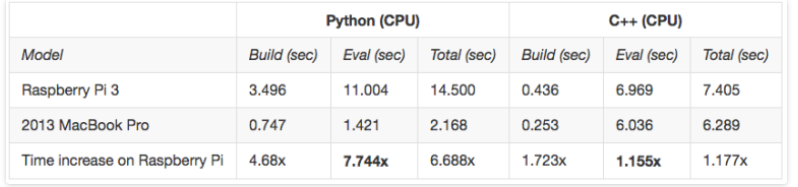
보통 Q테이블을 배열로 설정하는데 이를 딥러닝 CNN으로 변경한 것이 DQN입니다. 그렇기 때문에 상태나 행동이 큰 집합도 잘 학습이 된다고 합니다.



[라즈베리파이에서 물체 인식을 할 때 벤치마크 정보]

(<https://medium.com/nanonets/how-to-easily-detect-objects-with-deep-learning-on-raspberrypi-225f29635c74>)

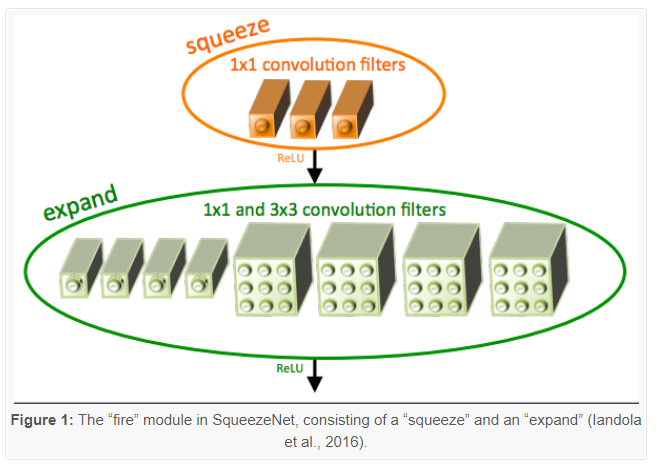
위 사진은 라즈베리파이 환경에서 양자화 후 해당 모델을 돌렸을 때의 성능 지표입니다.



[라즈베리파이와 맥북프로의 딥러닝 활용 비교]

Python에 비해 C++은 라즈베리파이와 맥북 프로의 차이가 크게 나지 않습니다.

라즈베리파이에서 인공 신경망을 사용하기 위해서는 작은 필터를 가진 모델을 선정해야 합니다 만약 학습된 모델을 사용한다면 GoogLeNet이나 SqueezeNet을 사용해야 할 것으로 보입니다. GoogLeNet은 25MB의 크기, SqueezeNet은 5MB의 크기입니다. 따라서 SqueezeNet이 GoogLeNet보다 더 빠른 이미지 판별을 할 수 있을 것으로 보이지만 정확성이 줄어드는 점을 주의해야할 것으로 보입니다.



[SqueezeNet]

**Reference**

AI Dev

<http://aidev.co.kr/deeplearning>

<https://tensorflow.blog/tag/raspberry-pi/>

pyimagesearch

<https://www.pyimagesearch.com/2017/10/02/deep-learning-on-the-raspberry-pi-with-opencv/>