

ch17_6_history_of_deep_learning

Tags

Deep Learning Basics

deep learning motivation

Perceptron

Perceptron 모델의 한계

Multi Layer Perceptron

Back Propagation

Back Propagation의 한계

Breakthrough

ImageNet

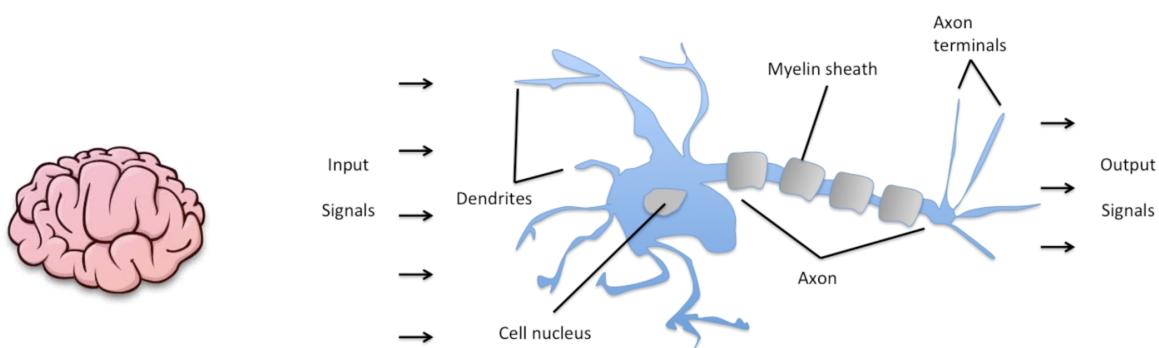
딥 러닝 발전의 배경

정리

Deep Learning Basics

deep learning motivation

Ultimate dream: thinking machine



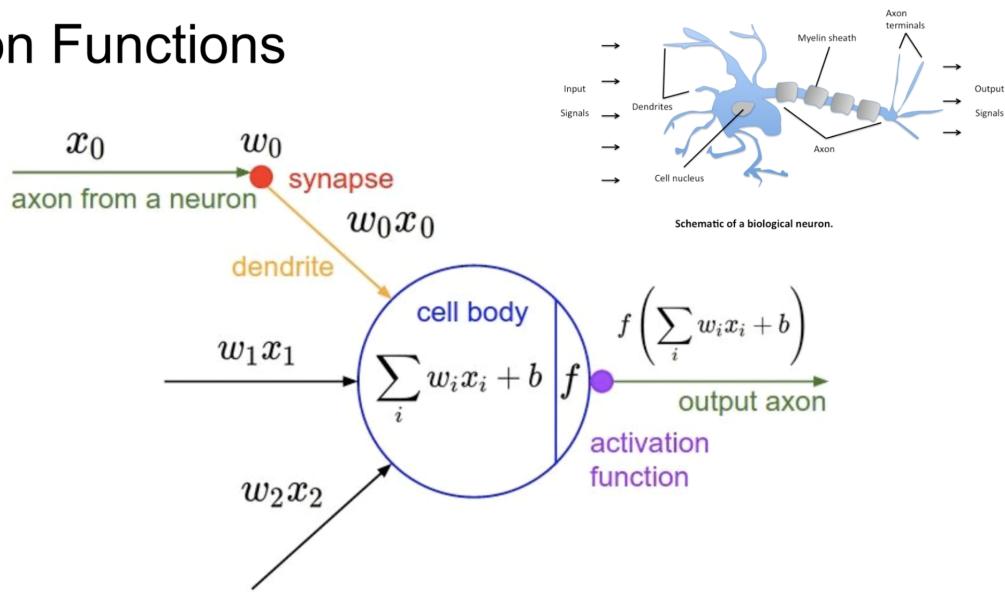
Schematic of a biological neuron.

AI 분야는 사람처럼 생각하는 알고리즘을 연구하는 분야입니다. 그 중에서도 딥러닝 우리 뇌가 동작하는 방식을 본따서 만든 알고리즘입니다.

우리 뇌의 뉴런들은 입력 신호를 받습니다. 이 신호들을 어떤 처리를 한 다음, 일정 크기 이상이면 활성화되고, 이후 뉴런들에 신호를 전달한다는 방식으로 동작합니다. 굉장히 단순하죠?

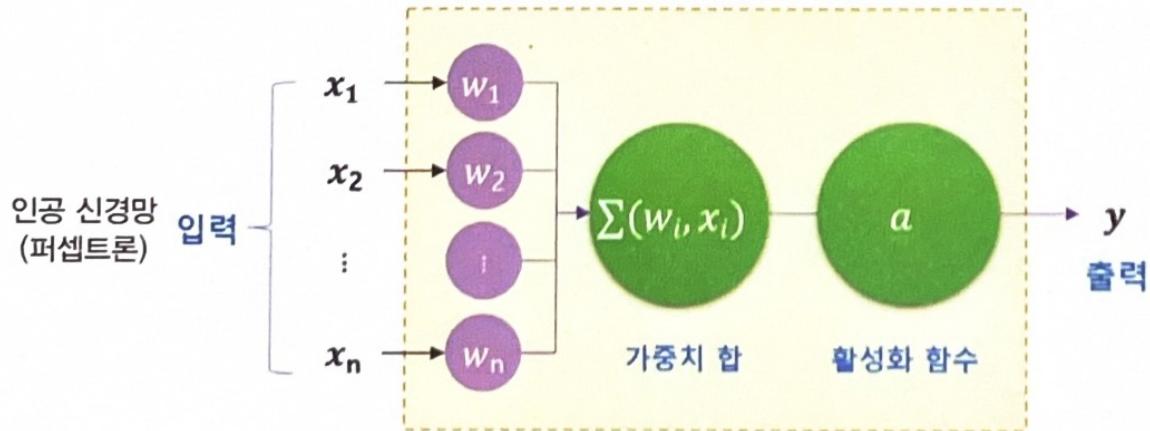
Perceptron

Activation Functions



딥러닝은 이런 뉴런 동작 방식을 본따서 만든 알고리즘입니다. 각각의 뉴런들은 linear model처럼 입력 신호에 대해서 weight를 더하고 bias를 더해줍니다. 이 값이 일정 수준보다 작으면 0 신호를 다음 뉴런에 전달하고, 일정 수준보다 크면 다음 뉴런에 그대로 신호를 전달합니다.

이렇게 신호를 입력받아서 일정한 처리를 하고, 그 결과 값의 크기에 따라서 다음 뉴런에게 전달하는 뉴런 하나를 perceptron이라고 부릅니다. 퍼셉트론 층 하나로 구성된 신경망 모델은 아래처럼 표현할 수 있습니다.

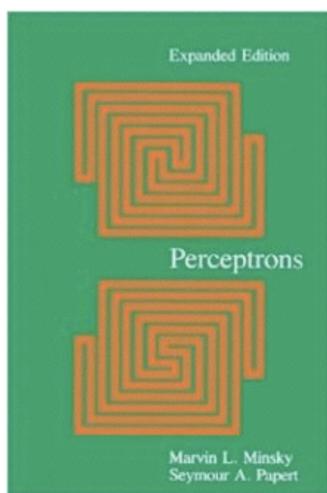


[그림 5-1] 신경세포 vs. 인공 신경망

Perceptron 모델의 한계

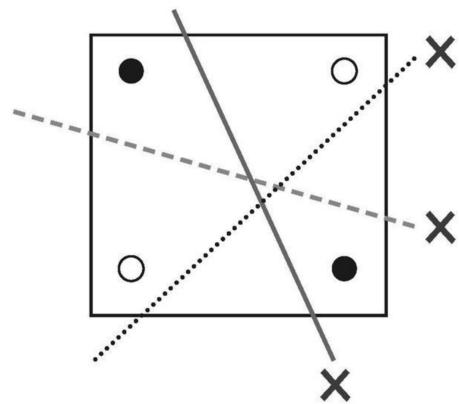
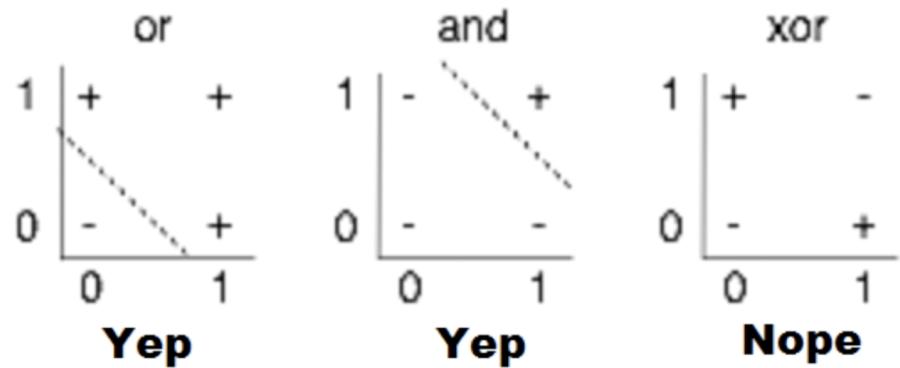
Perceptrons (1969)

by Marvin Minsky, founder of the MIT AI Lab



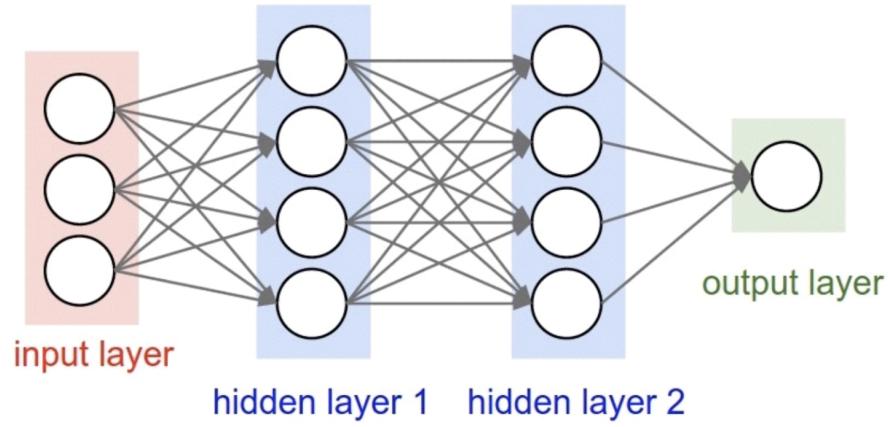
- We need to use MLP, multilayer perceptrons (multilayer neural nets)
- No one on earth had found a viable way to train MLPs good enough to learn such simple functions.

당시 MIT AI LAB의 민스키 교수가 Perceptrons라는 저서에서 perceptron 모델만으로는 XOR 연산을 구현할 수 없다고 비판하게 됩니다. 쉽게 얘기하면 or나 and 연산은 선형 모델로 구현이 가능한데, XOR 연산은 불가능 하다는 것입니다.



Multi Layer Perceptron

그러면서 민스키 교수는 xor 연산을 수행할 수 있는 모델을 만들려면 퍼셉트론을 여러개 쌓은 Multi Layer Perceptron 모델이 필요하다고 말했습니다. 이는 퍼셉트론을 여러개 쌓아서 만든 모델로 줄여서 MLP라고 부릅니다.



(MLP 모델이 어떻게 XOR을 구현할 수 있는지 궁금하신 분들은 아래 링크를 참고하세요)

<https://ang-love-chang.tistory.com/26>

문제는 이 multilayer perceptron 모델을 학습시킬 방법이 당시엔 없었다는 점입니다. 민스키 교수는 지구상 그 누구도 multilayer perceptron 모델을 학습시킬 수 있는 방법을 찾지 못했다고 저서에 못을 콤 박아버립니다. 이는 당시의 AI 연구에 찬물을 끼얹고 투자와 연구가 크게 감소했습니다. 이 시기를 첫번째 AI winter라고 부릅니다.

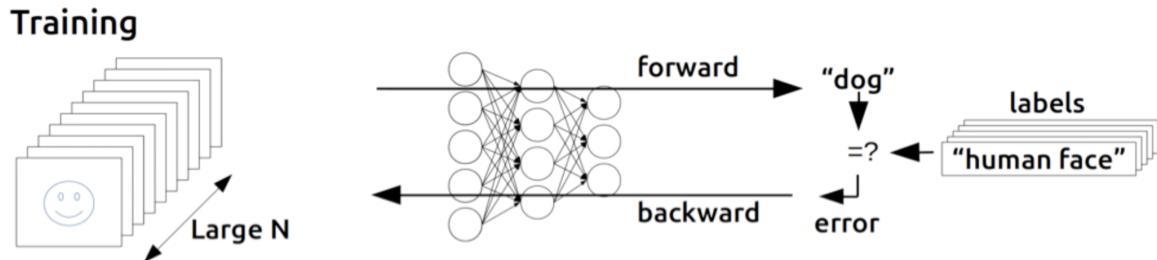
(AI winter에 대해서 더 궁금하신 분들은 아래 링크를 참고해보세요)

<https://the14thfloor.tistory.com/150>

Back Propagation

Backpropagation

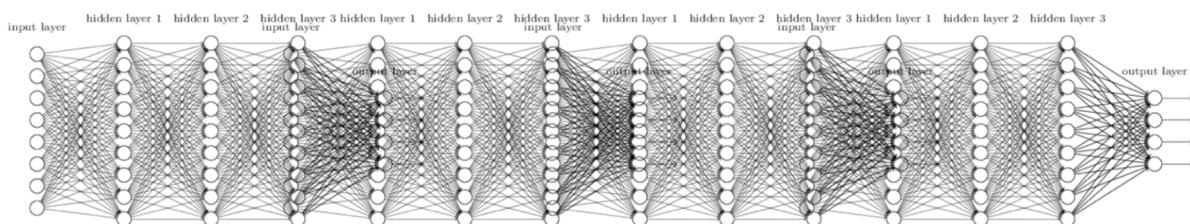
(1974, 1982 by Paul Werbos, 1986 by Hinton)



MLP를 학습시키지 못한다는 한계점은 오류 역전파 알고리즘으로 극복됩니다. 신경망을 통해서 예측 값을 얻은 다음, 에러를 계산해서 이를 모델에 역전파를 시켜서 MLP를 학습시키자는 아이디어입니다. 이는 폴에 의해서 1974년 제안되었으나 당시에는 주목을 받지 못했습니다. 이후에 제프리 힌튼 교수에 의해서 1986년도에 다시금 발표되었고, 그 때부터 주목을 받기 시작합니다. (힌튼 교수님은 딥러닝의 아버지라고도 불립니다.)

힌튼 교수님 근황: <https://www.bbc.com/korean/articles/crgm8d787l7o>

Back Propagation의 한계



MLP의 학습을 가능하게 해주었던 Back Propagation 알고리즘은 층이 몇개 안되는 간단한 모델은 충분히 학습시킬 수 있었습니다.

그러나 복잡한 문제를 풀기 위해서 신경망을 깊게 쌓으면 오류 역전파 시에 출력 층으로부터 멀리 떨어진 층까지 충분히 전달되지 않는다는 문제가 있었습니다. 이를 vanishing gradient라고 부릅니다.

동시에 SVM이나 RandomForest와 같은 단순하면서도 잘 동작하는 알고리즘들이 등장하면서 AI 연구는 두번째 겨울을 맞습니다.

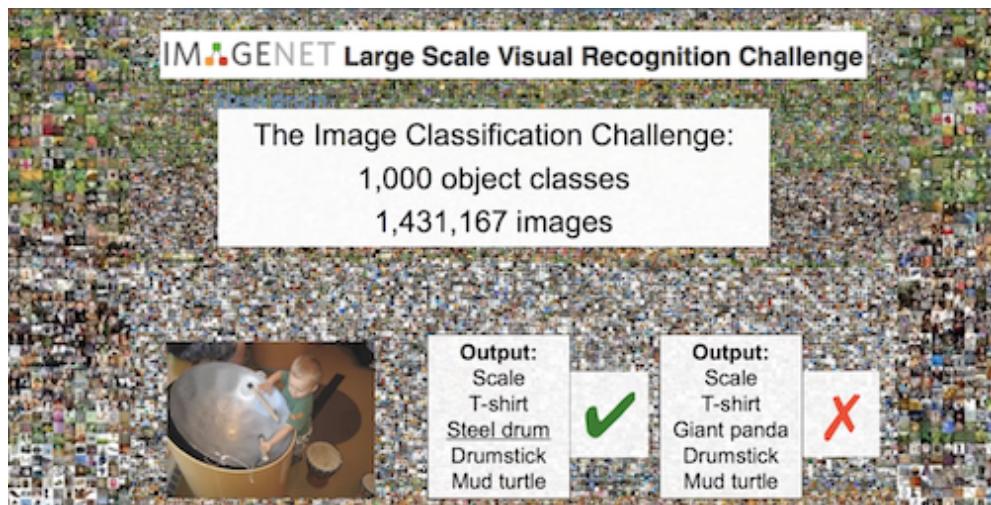
Breakthrough

두번째 AI winter 기간에 뉴럴 네트워크 연구에 대한 지원이 대폭 감소하였습니다. 당시에 힌튼 교수님은 거의 유일하게 AI 연구를 지원해주는 캐나다의 연구 기관인 CIFAR의 지원을 받아 캐나다로 이주합니다.

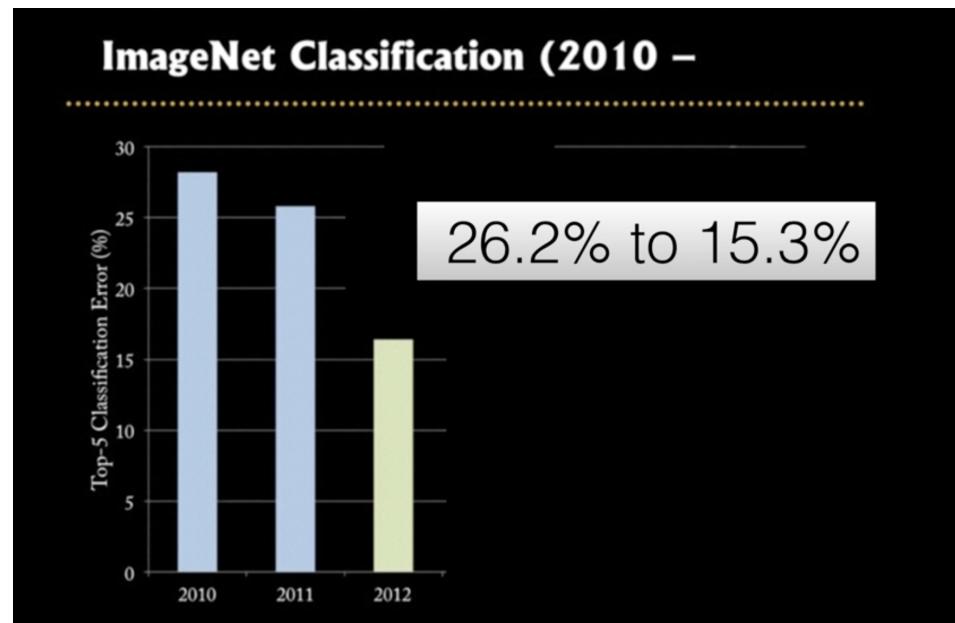
이후 연구를 거듭하여 2006년과 2007년, 힌튼 교수와 벤지오 교수가 기념비적인 논문을 발표합니다. 하나는 딥 뉴럴 네트워크의 퍼셉트론들의 초기값만 잘 설정하더라도 학습이 효과적으로 일어난다는 것이고, 다른 하나는 깊은 신경망이 복잡한 문제를 효과적으로 풀어낸다는 내용이었습니다.

그러면서 neural network라는 이름을 deep network, deep learning으로 브랜딩 하면서 다시금 사람들의 주목을 받게 됩니다.

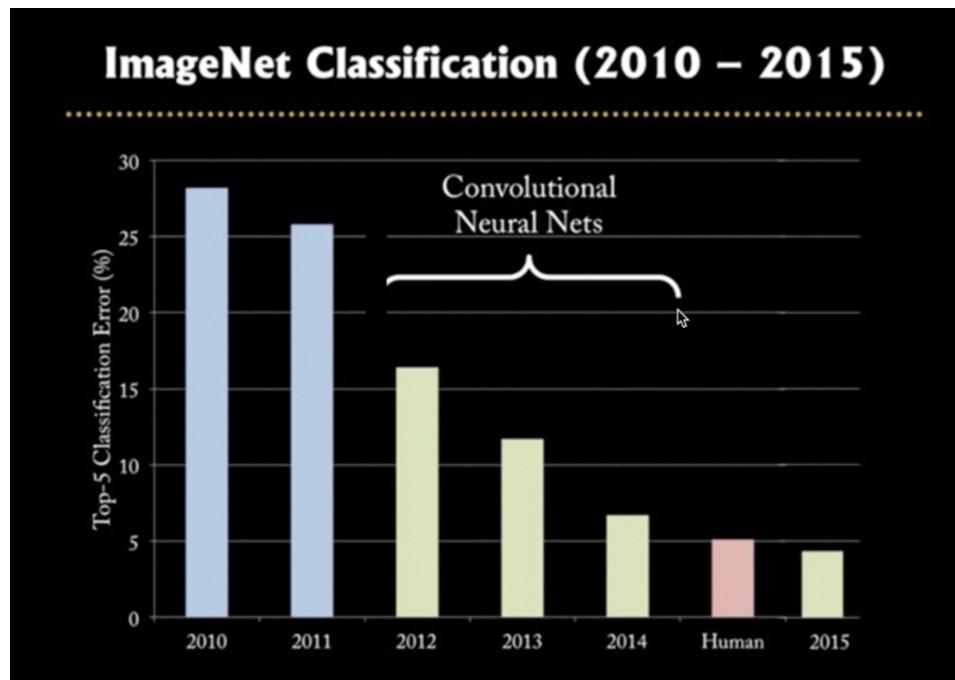
ImageNet



딥 러닝이 본격적으로 주목을 받게된 계기는 ImageNet Challenge였습니다. 이미지넷 챌린지는 1000개의 종류의 물체에 대해서 수집한 140만장의 이미지에 대해서 분류를 수행하는 대회였습니다.



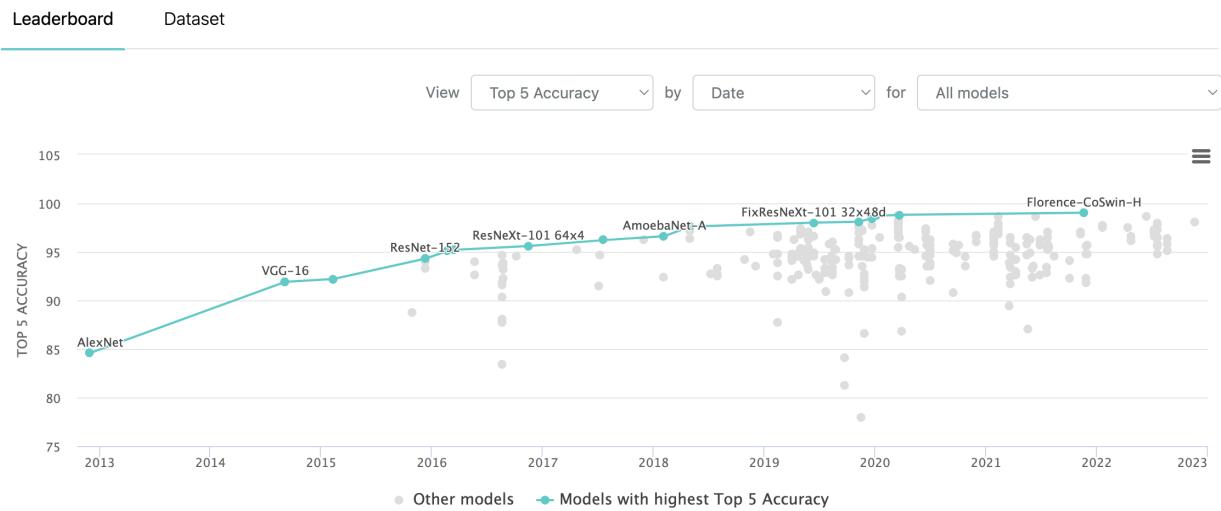
지금은 간단해보이는 테스크이지만, 불과 2010년대만 하더라도 오류율이 30% 가까이 되었습니다. (top 5 accuracy 기준) 또한 당시에 사용했던 컴퓨터 비전 알고리즘은 수학적으로 너무 복잡하였습니다. 이런 상황에서 힌튼 교수님 연구실의 대학원생이 딥러닝 기반 모델을 공개하여 26%였던 기존의 오류를 15.3%로 줄이게 됩니다.



딥러닝 기반의 모델은 충격적으로 성능이 뛰어남과 동시에 구조적으로 매우 단순했습니다. 많은 연구자들이 딥러닝 연구에 뛰어들었고, 빠른 속도로 발전하면서 급속도로 성능이 향상되어 심지어 사람보다도 이미지 분류를 잘해내는 수준에 이르렀습니다.

top5 accuracy

Image Classification on ImageNet



top1 accuracy

Image Classification on ImageNet



이제는 단순 이미지 분류는 너무 쉬운 테스크라서 학계의 주목을 덜 받고 있기는 하지만, 꾸준히 논문들이 발표되면서 top 5 accuracy 기준으로는 99.02%, top1 accuracy 기준으로는 90.05%의 정확도를 기록합니다. 90%라고 해서 애개? 라고 하실 수도 있지만, 실제로 오답들을 살펴보면 이미지 하나에 여러 물체가 포함되어 있어서 정답을 하나만 고르기 애매한 이미지들이 대부분입니다.

딥 러닝 발전의 배경

- 대량의 데이터셋 (imagenet)
- 컴퓨팅 파워의 발전 (GPU)
- 비선형 함수 사용 (ReLU)
- 웨이트 초기화 기법의 개선

정리

이번 챕터에서는 딥러닝의 발전 과정과 역사, 핵심적인 개념들에 대해서 짚어보았습니다. 이제 본격적으로 torch를 이용해서 딥 뉴럴 네트워크를 코딩해보겠습니다.

