# KNOWLEDGE DISTILLATION

## IN A NEURAL NETWORK

# REFERENCES


- Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531 (2015).
- https://www.ttic.edu/dl/dark14.pdf

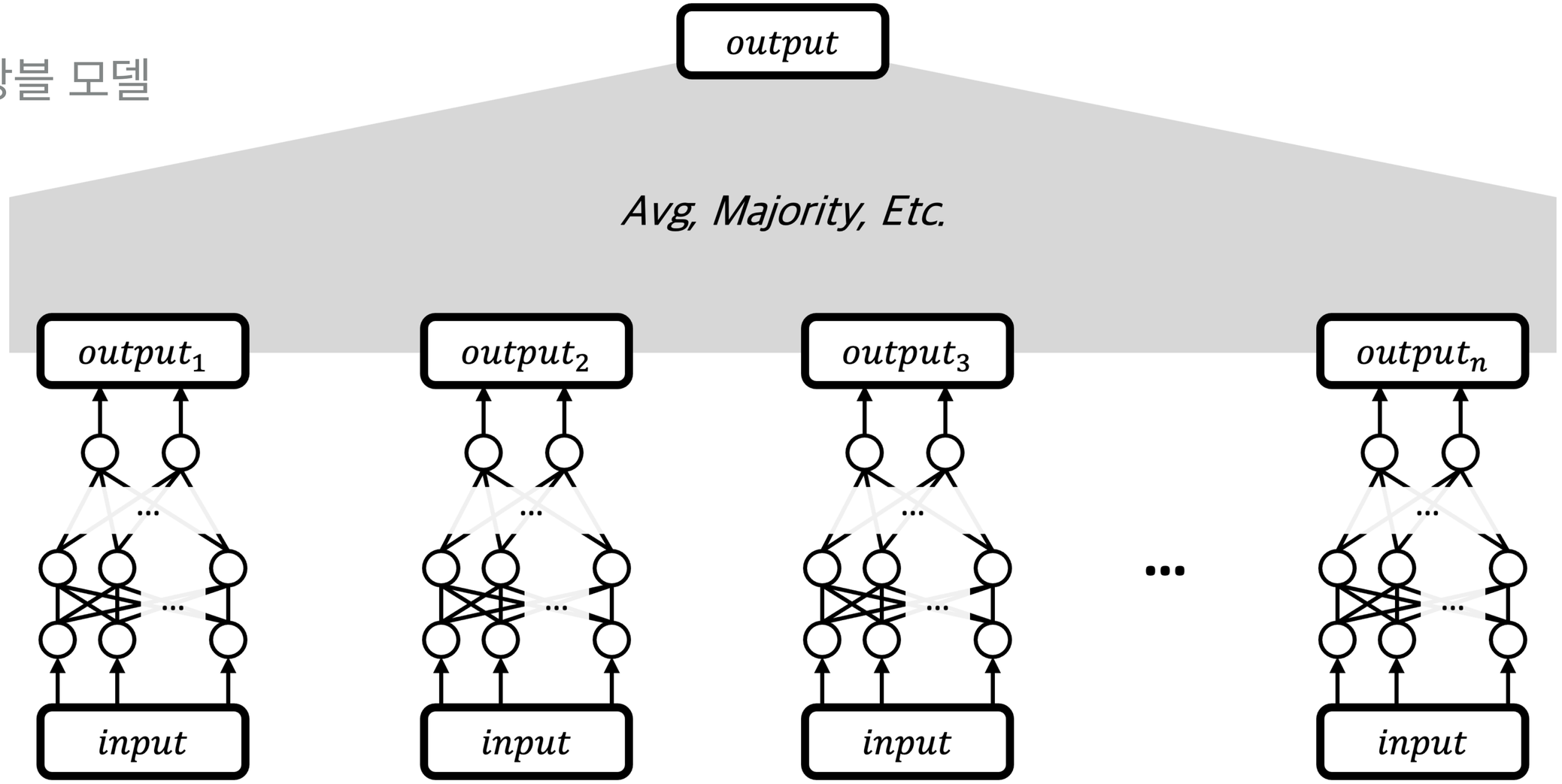
# INTRODUCTION

# INTRODUCTION

- 머신 러닝의 성능을 높이는 가장 쉬운 방법은
  - 서로 다른 여러 모델을 만들고
  - 예측 결과를 평균 내는 것

# INTRODUCTION

- 앙상블 모델



# INTRODUCTION

- 그러나 이러한 앙상블 방법은 비용이 높음
  - 각 신경망이 큰 모델일 경우 더 심함

# INTRODUCTION

- 앙상블 모델을 하나의 싱글 모델로 지식 증류를 한다면
  - 다루기 훨씬 쉬워질 것
- 어떻게 할 수 있을까?
- 잘 될 것인가?

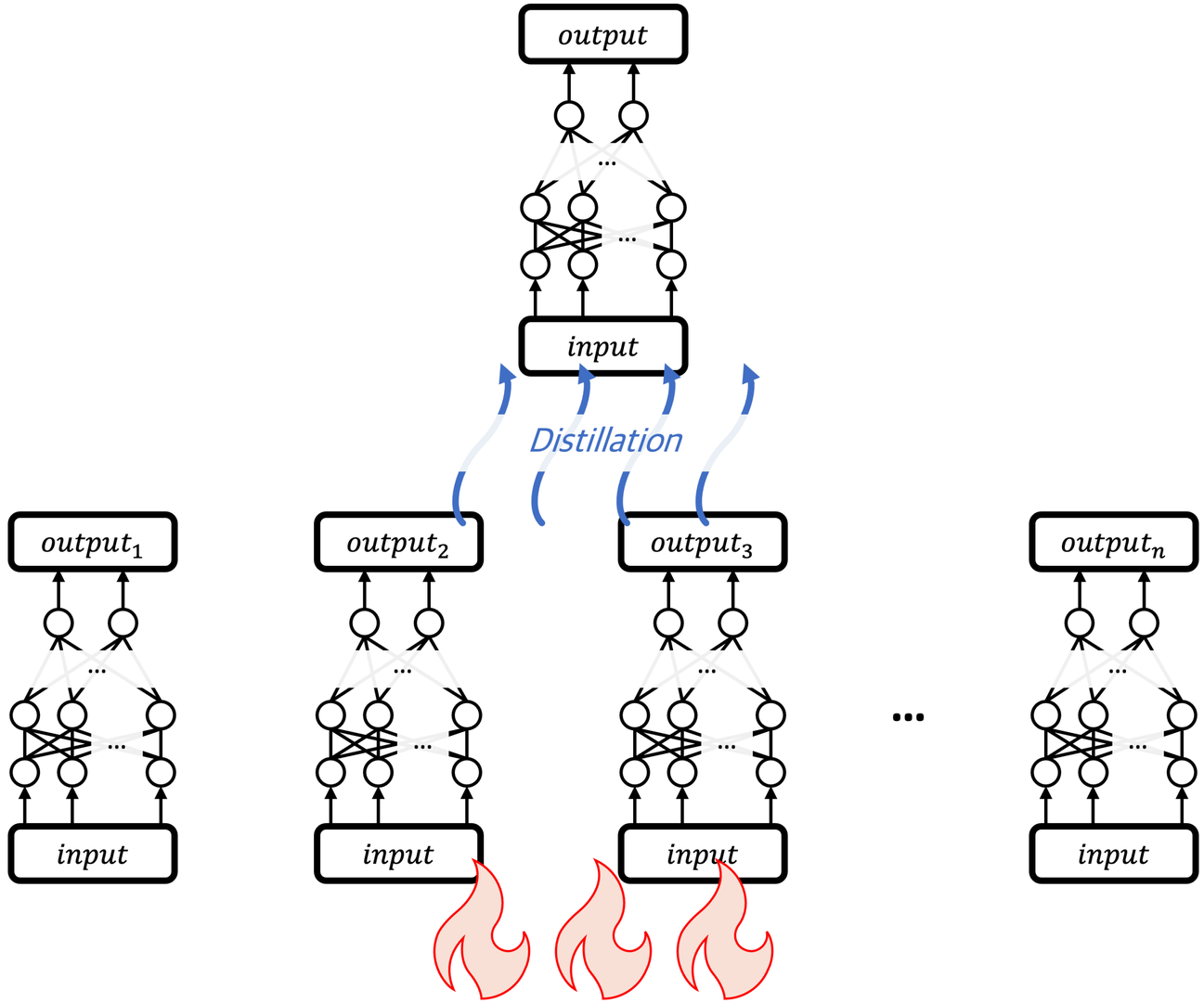
# DISTILLATION

# DISTILLATION

- 증류
  - 혼합물에서 특정 성분을 분리시키는 방법
- 신경망 증류
  - 복잡한 모델의 일반화 능력을 전달하자

# DISTILLATION

- 신경망 증류



# DISTILLATION

- 이미지 분류의 예시
  - 최종 output은 소프트맥스(softmax)의 형태
  - 출력은 0~1 사이
  - 출력의 합이 1
- 이 소프트맥스 레이어의 값이 모델의 지식에 해당
  - 앙상블 모델의 소프트맥스 출력을 전달하자
  - “지식”을 배우자!

# DISTILLATION

- 소프트맥스 출력의 값
  - 특정 범주의 값이 0에 매우 가까울 수 있음
  - 지식이 잘 전달되지 않음

# DISTILLATION

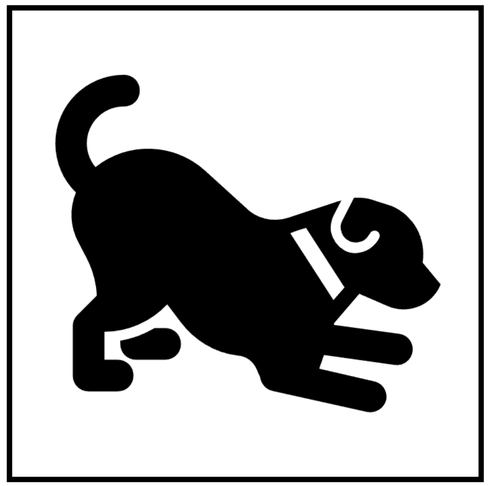
- Softened output of softmax
  - $p_i = \frac{exp(z_i/T)}{\Sigma_j exp(z_j/T)}$
  - 기존 소프트맥스 함수에 인자 $T$ 추가
  - 온도 $T$ 가 높을수록 soft 해짐
    - 온도가 높을수록 증류가 잘 됨

# DISTILLATION

- Softened output of softmax
  - $p_i = \frac{exp(z_i/T)}{\Sigma_j exp(z_j/T)}$
  - $T$ 가 1이면 통상의 소프트맥스 함수
  - 실험적으로 $T$ 가 2~4 정도에서 증류가 효과적

# DISTILLATION

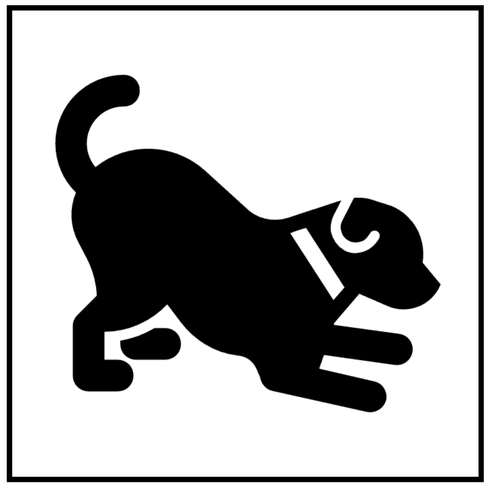
- 하드 레이블, 앙상블 출력, softened 앙상블 출력



| bird | car | dog | ⋯ | cat | |
|---|---|---|---|---|---|
| 0 | 0 | 1 | | 0 | *Hard label* |
| $10^{-8}$ | $10^{-6}$ | 0.85 | | 0.1 | *Output of Ensemble* |
| 0.005 | 0.07 | 0.3 | | 0.2 | *Softended Output of Ensemble* |

# DISTILLATION

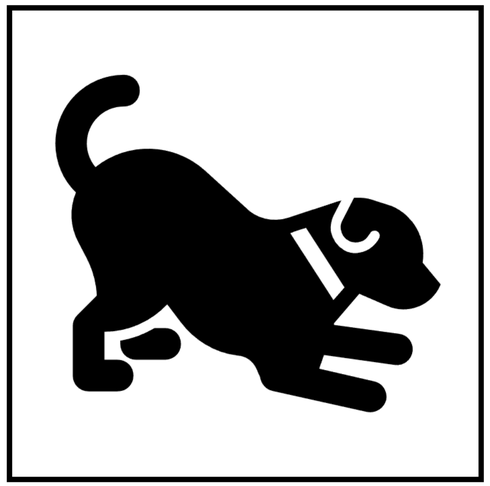
- 하드 레이블, 앙상블 출력, softened 앙상블 출력



| bird | car | dog | … | cat | |
|---|---|---|---|---|---|
| 0 | 0 | 1 | | 0 | *Hard label* |
| $10^{-8}$ | $10^{-6}$ | 0.85 | | 0.1 | = 지식 *Output of Ensemble* |
| 0.005 | 0.07 | 0.3 | | 0.2 | *Softended Output of Ensemble* |

# DISTILLATION

- 하드 레이블, 앙상블 출력, softened 앙상블 출력



| bird | car | dog | ⋯ | cat | |
|---|---|---|---|---|---|
| 0 | 0 | 1 | | 0 | *Hard label* |
| $10^{-8}$ | $10^{-6}$ | 0.85 | | 0.1 | *Output of Ensemble* |
| 0.005 | 0.07 | 0.3 | | 0.2 | *Softended Output of Ensemble* |

= Dark knowledge

# TRAINING

# TRAINING

- 지식을 배우자!
  - 선생(Teacher, T) 모델이 학생(Student, S) 모델에게 지식을 전달
- 큰 모델인 T를 학습시킨 후
- 작은 모델인 S를 다음 손실함수로 학습:
  - $$L = \sum_{(x,y)\in\mathbb{D}} \lambda_1 * L_{KD}(S(x, \theta_S, \tau), T(x, \theta_T, \tau)) + \lambda_2 * L_{CE}(\hat{y}_S, y)$$

# TRAINING

- 작은 모델인 S를 다음 손실함수로 학습:
  - $$L = \sum_{(x,y)\in\mathbb{D}} \lambda_1 * L_{KD}(S(x, \theta_S, \tau), T(x, \theta_T, \tau)) + \lambda_2 * L_{CE}(\hat{y}_S, y)$$
  - 모든 (데이터, 레이블) 의 쌍 $(x, y)$ 에 대해

# TRAINING

- 작은 모델인 S를 다음 손실함수로 학습:
- $$L = \sum_{(x,y)\in\mathbb{D}} \lambda_1 * L_{KD}(S(x, \theta_S, \tau), T(x, \theta_T, \tau)) + \lambda_2 * L_{CE}(\hat{y}_S, y)$$
- Distillation Loss
  - 온도 $\tau$ 로 구한 지식의 KD (Knowledge Distillation)
  - S의 soft prediction, T의 soft label의 차이
  - 계수 $\lambda_1$
  - Cross Entropy Loss 사용

# TRAINING

- 작은 모델인 S를 다음 손실함수로 학습:
  - $$L = \sum_{(x,y)\in\mathbb{D}} \lambda_1 * L_{KD}(S(x, \theta_S, \tau), T(x, \theta_T, \tau)) + \lambda_2 * L_{CE}(\hat{y}_S, y)$$
- Cross Entropy Loss
  - 일반 신경망 학습에 사용하는 손실과 동일
  - S의 (hard) prediction, Data의 hard label의 차이
  - 계수 $\lambda_2$

# EXPERIMENT

# EXPERIMENT ON MNIST - 1

- 기본 모델
  - 784 —> 800 —> 800 —> 10 구조의 신경망 A
  - 146 test error

# EXPERIMENT ON MNIST - 1

- 확장된 모델
  - 784 —> 1200 —> 1200 —> 10 구조의 신경망 B
  - Dropout 등 적용
    - 앙상블의 간소화 버전
  - 67 test error

# EXPERIMENT ON MNIST – 1

- A 와 동일한 구조의 신경망 C
  - KD를 통해 학습
  - 74 test error
    - 신경망 A의 146 대비 큰 절감

# EXPERIMENT ON MNIST – 2

- 지식을 통해 학습하지 않은 범주를 유추할 수 있을까?
- “3” 범주 없이 학습
- “3”에 대해서는 오직 KD를 통해서만 간접 학습

# EXPERIMENT ON MNIST – 2

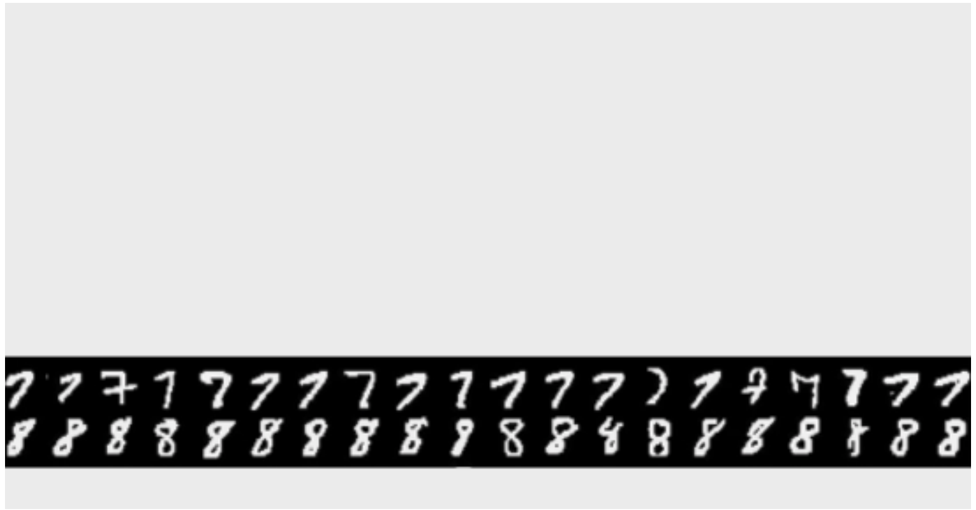
- 결과
  - 오직 109 test error
  - 1010개의 “3” 중에 14개만 틀림!
    - 98.6% 정확도

# EXPERIMENT ON MNIST – 2

- 학습 과정에서 “3”을 본 적이 없지만
- 지식을 통해 이를 유추
  - 숫자 “2”가 “3”이랑 얼마나 비슷하고,
  - 숫자 “5”는 “3”이랑 얼마나 비슷하고, …
- 실제로 “3”을 만났을 때
  - “3”이라는 사실을 알아챌 수 있음

# EXPERIMENT ON MNIST – 3

- 지식을 통해 학습하지 않은 범주를 유추할 수 있을까?
- 대부분의 범주를 다 생략해보자
- “7”과 “8” 외 나머지 전부 숨김



# EXPERIMENT ON MNIST – 3

- 모든 범주에 대해 87%의 정확도
- KD를 통해
  - 학습 데이터의 전체가 아닌
  - 편향된 일부분으로도
  - 대부분의 정보를 학습할 수 있음

# CONCLUSION

# CONCLUSION

- 앙상블 또는 큰 모델의 지식을 작은 네트워크에 전달
- 소프트맥스 함수의 값을 이용해 지식 증류를 할 수 있음
  - Hard Label을 쓸 때 보다 많은 정보가 함축되어 있음
  - 온도를 이용해 soften 정도를 조절할 수 있음
- 실제 딥러닝 서비스에서 유용하게 활용 가능

# KNOWLEDGE DISTILLATION

## IN A NEURAL NETWORK