Computer Vision (2주차)

Young-Gon Kim
DLI Instructor

DEEP LEARNING INSTITUTE

DLI Mission

Helping people solve challenging problems using AI and deep learning.

- Developers, data scientists and engineers
- Self-driving cars, healthcare and robotics
- Training, optimizing, and deploying deep neural networks

TOPICS

- Week 1 Review
- Overfitting
- Big Data
- Transfer Learning

WEEK 1 REVIEW

OVERFITTING

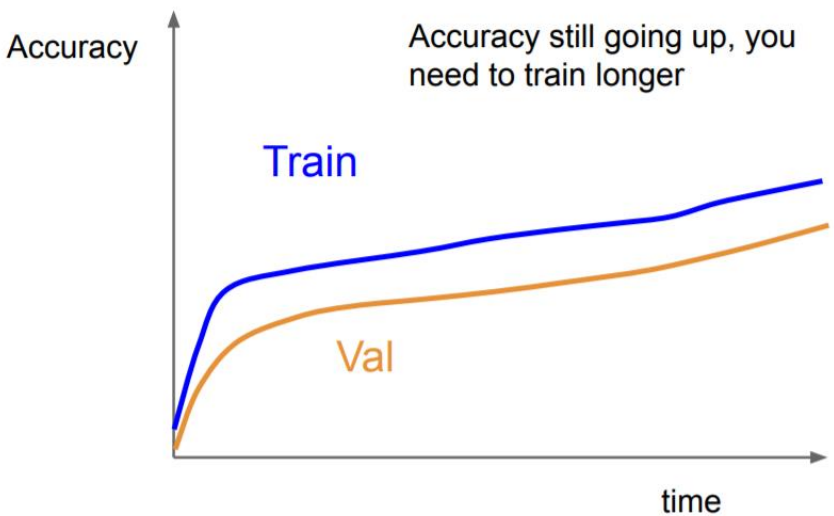
OVERFITTING

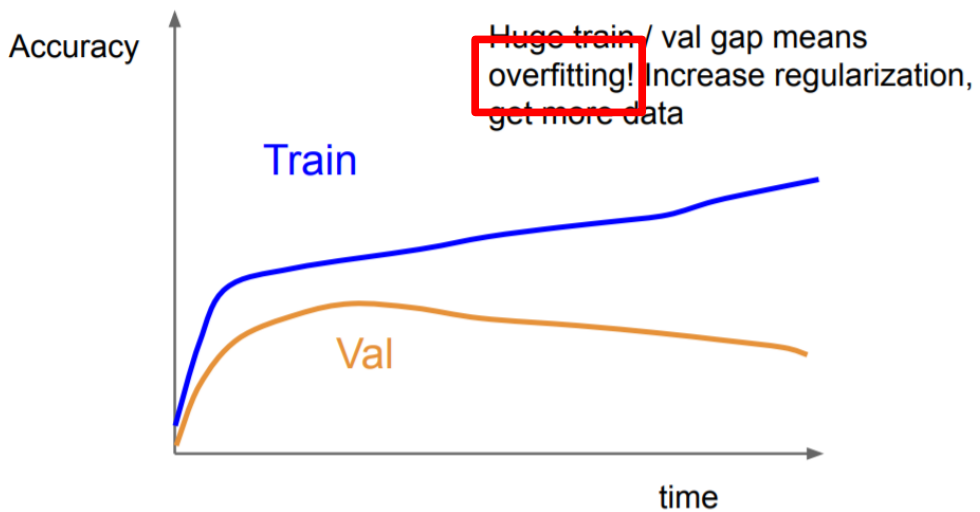
Louie! Image Classification Model

Predictions

Not Louie	96.52%
Louie	3.48%

OVERFITTING





http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture08.pdf

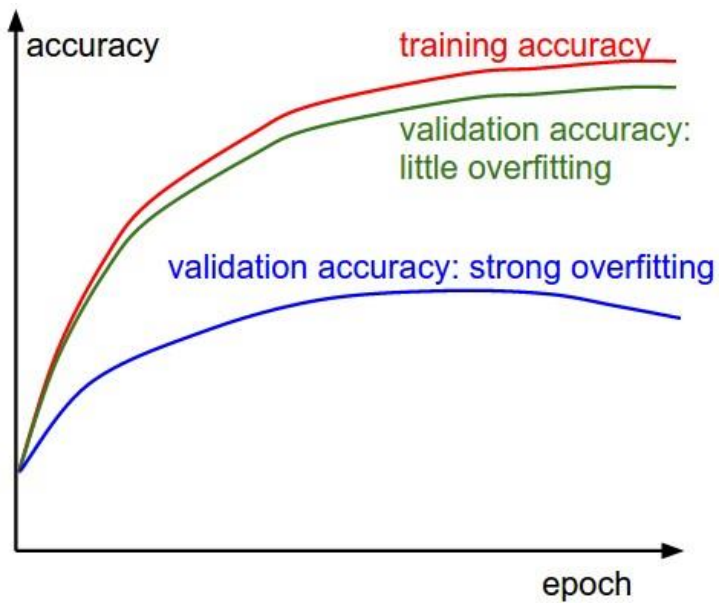
OVERFITTING

- Type of ‘overfitting’
  - **Green :** little overfitting

    → Model capacity is not high enough!

  - **Blue :** strong overfitting

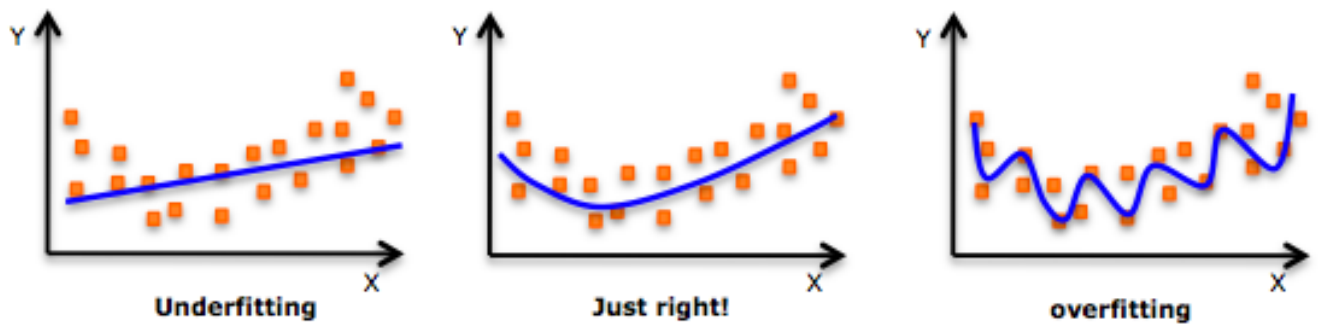
    → Model’s complexity is too high!



http://cs231n.github.io/neural-networks-3/

OVERFITTING

- How to solve the ‘overfitting’?
  1. **Regularization**
  2. **Early stopping**
  3. **Dropout**



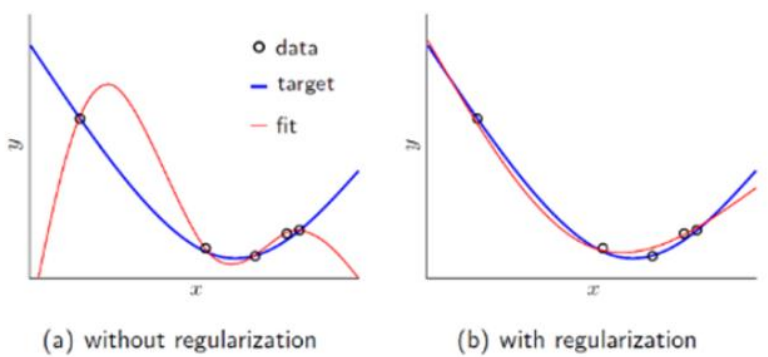
http://cs231n.github.io/neural-networks-3/

OVERFITTING

- How to solve the ‘overfitting’?
  1. **Regularization**
     - Loss function에 **penalty term**을 추가 (L1/L2 regularization)
     - **Variance를 낮춰서** model의 complexity를 낮춤 (**smoothing**)
  2. Early stopping
  3. Dropout

$$L1 \text{ regularization} = \frac{1}{n}\sum_{i=1}^{n}\{L(y_i, \hat{y}_i) + \frac{\boldsymbol{\lambda}}{2}|\boldsymbol{w}|\}$$

$$L2 \text{ regularization} = \frac{1}{n}\sum_{i=1}^{n}\{L(y_i, \hat{y}_i) + \frac{\boldsymbol{\lambda}}{2}|\boldsymbol{w}|^2\}$$

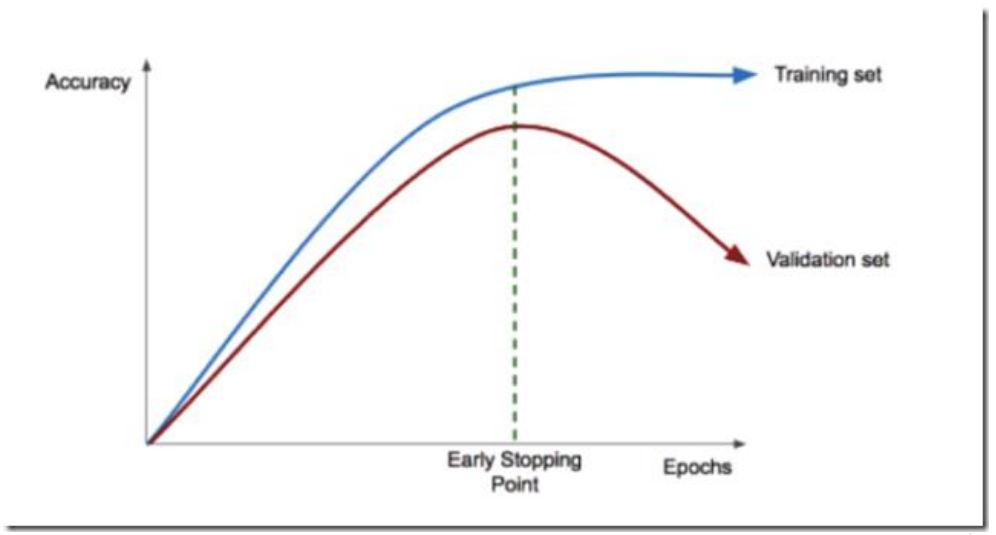


(a) without regularization (b) with regularization

https://nittaku.tistory.com/289
https://laonple.blog.me/220527647084

OVERFITTING

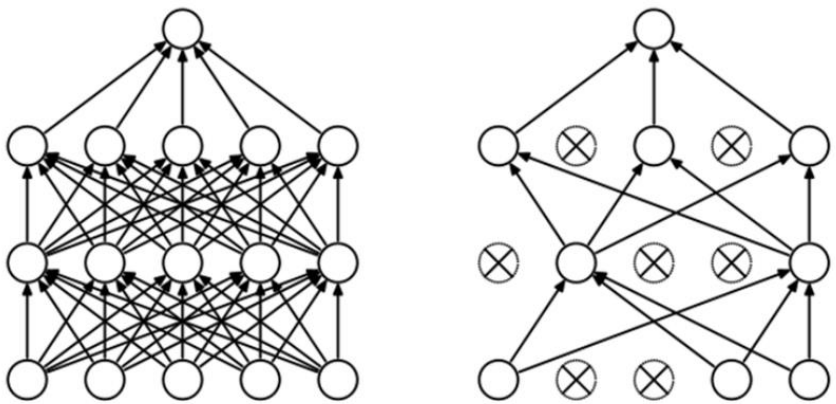
- How to solve the ‘overfitting’?
  1. Regularization
  2. **Early stopping**
     - Validation set accuracy가 멈추거나 낮아지는 지점 존재
     - 이 시점에서 **학습 중지**
  3. Dropout



https://nittaku.tistory.com/289

OVERFITTING

- How to solve the ‘overfitting’?
  1. Regularization
  2. Early stopping
  3. **Dropout**
     - Network의 **일부를 생략하고 학습**
     - Model ensemble 효과



(a) Standard Neural Net

(b) After applying dropout.

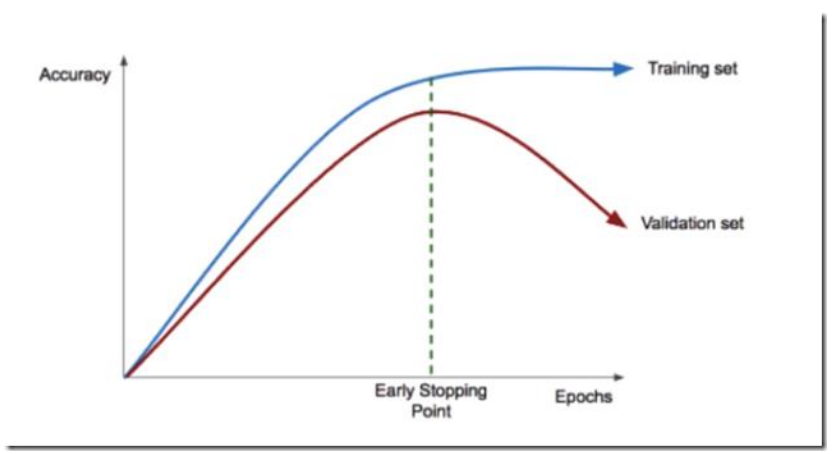
https://laonple.blog.me/220542170499

Training, Validation, Test

DATASET SPLIT

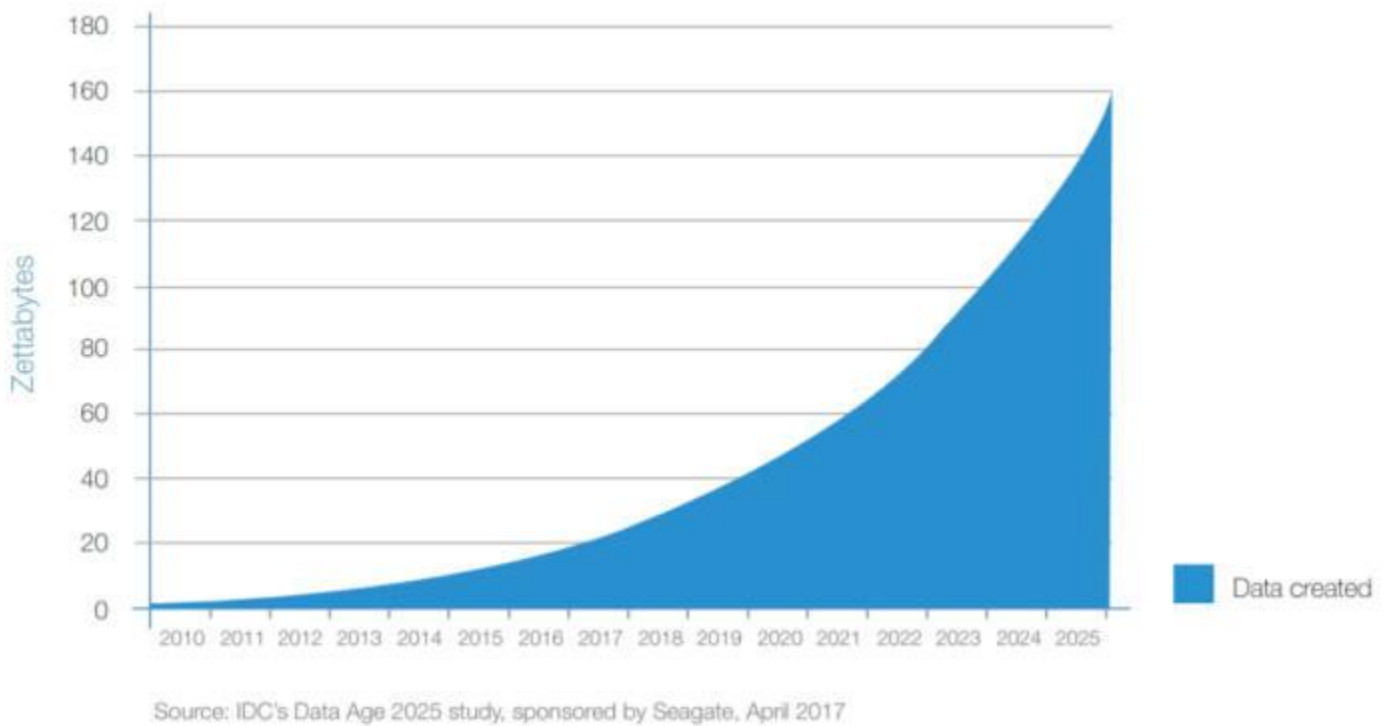
A Total of dataset

Training	Validation	Test



BIG DATA

BIG DATA



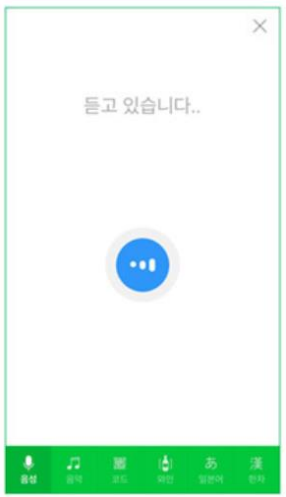
https://m.etnews.com/20171128000218

BIG DATA

- Big data와 Deep learning
  - **방대한 데이터를 다룰 수 있는 기술**의 발전

    → **Deep learning의 발전**에 큰 역할

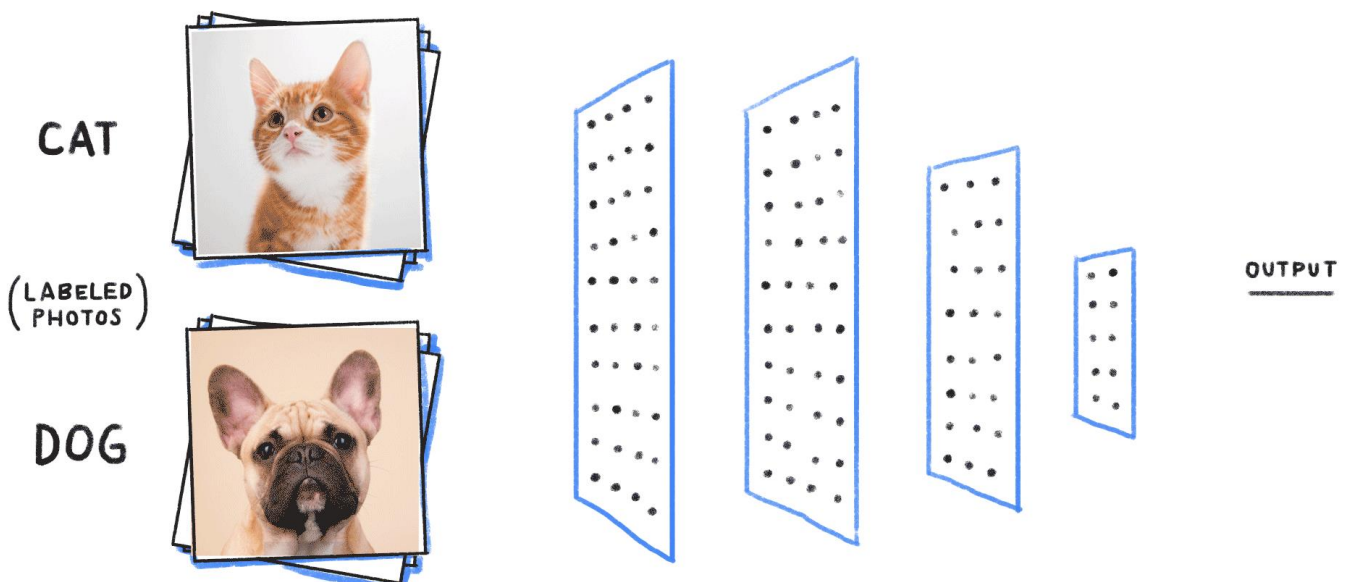
    → 우리 생활에서 큰 역할을 하기 시작



https://www.samsungsds.com/global/ko/support/insights/1196843_2284.html

BIG DATA

- Examples about deep learning using big data
  → Dog & Cat classification



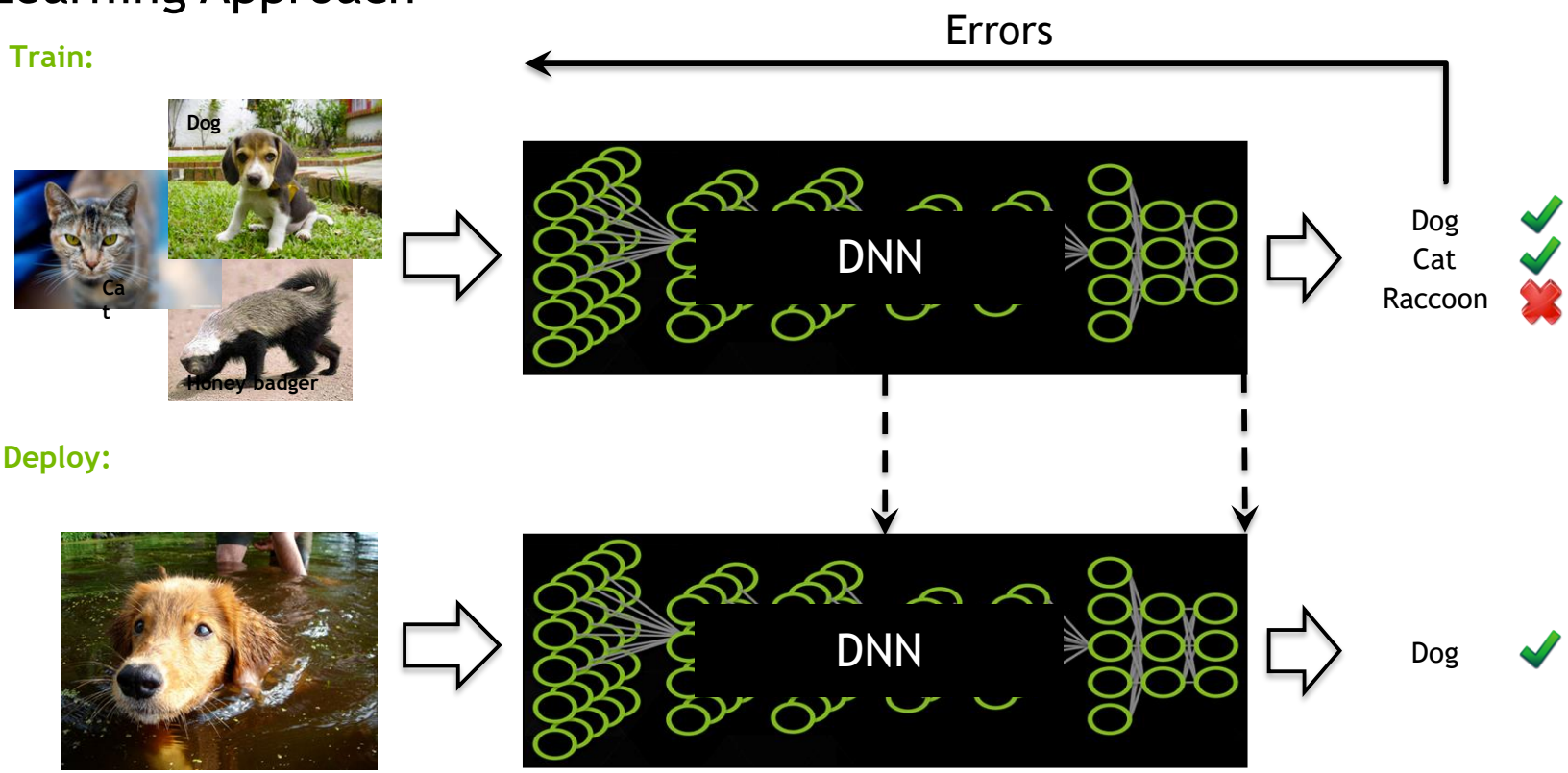
https://becominghuman.ai/building-an-image-classifier-using-deep-learning-in-python-totally-from-a-beginners-perspective-be8dbaf22dd8

Task 2
BIG DATA

DEPLOYMENT

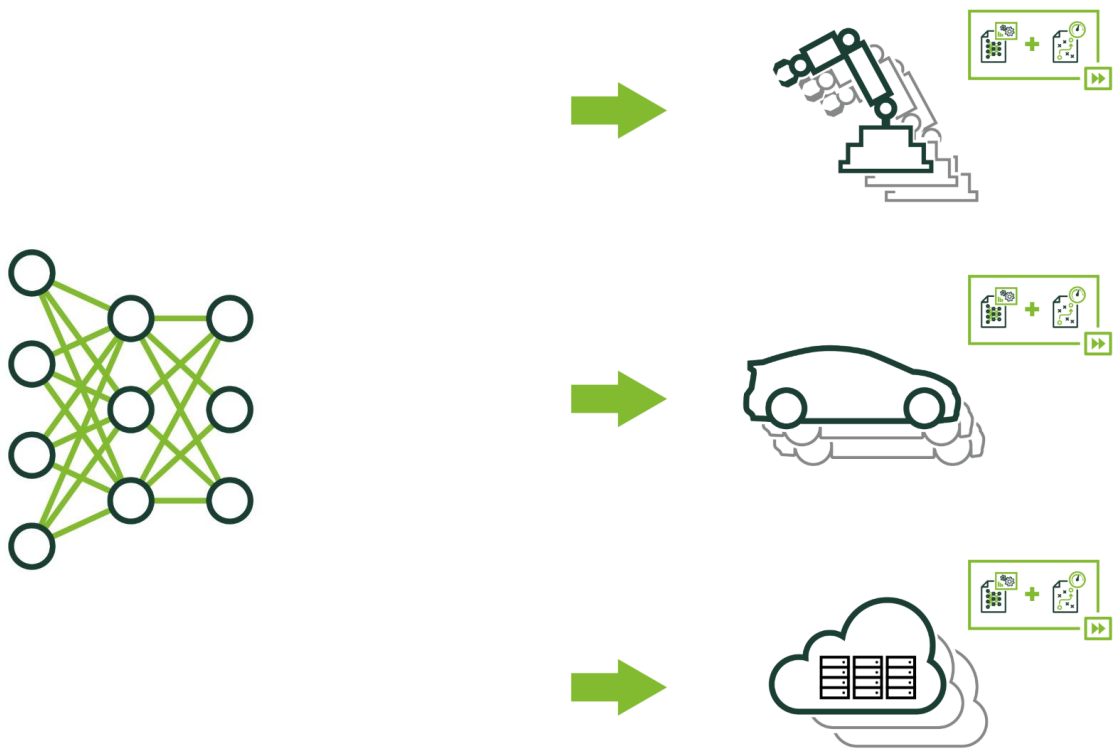
DEPLOYMENT

- Deep Learning Approach



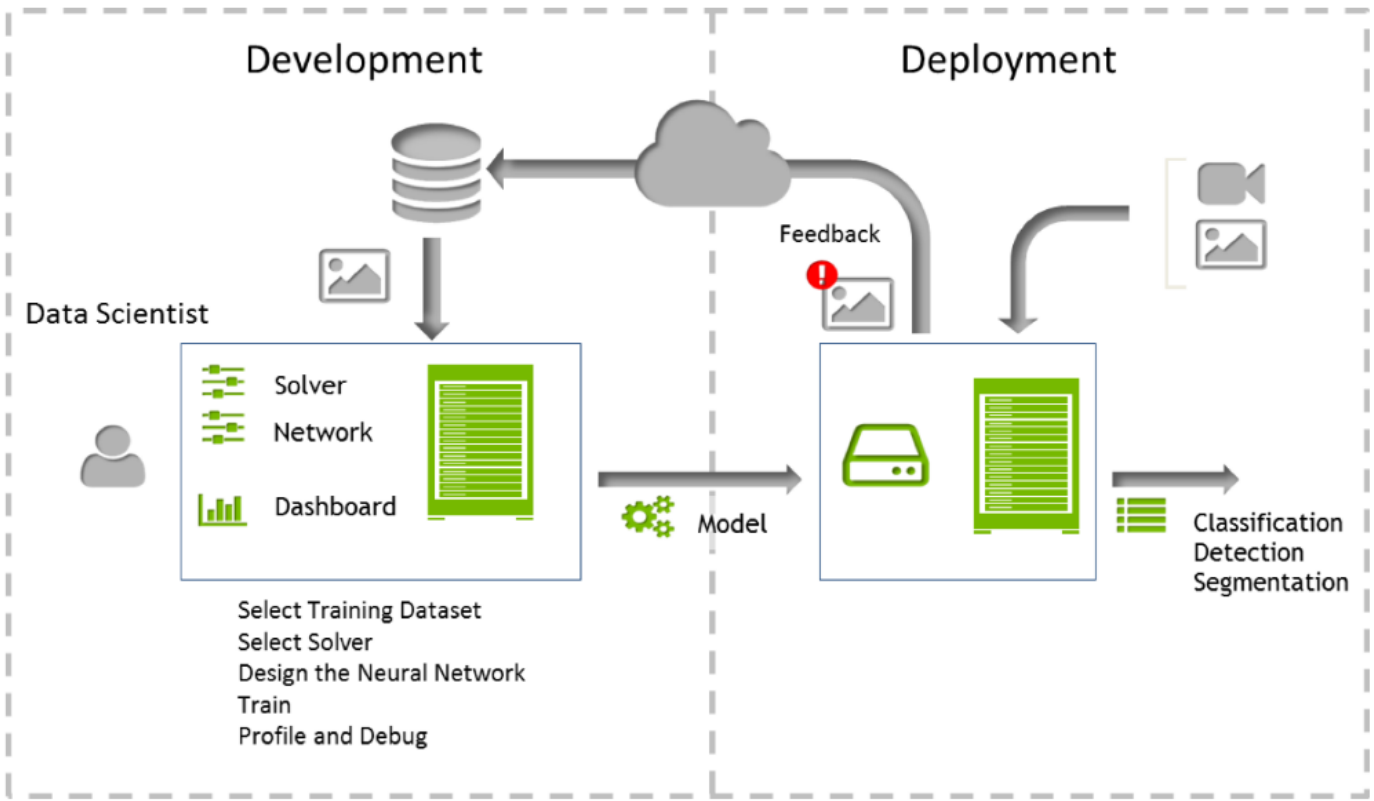
DEPLOYMENT

- How do I use a trained neural network as part of a solution?



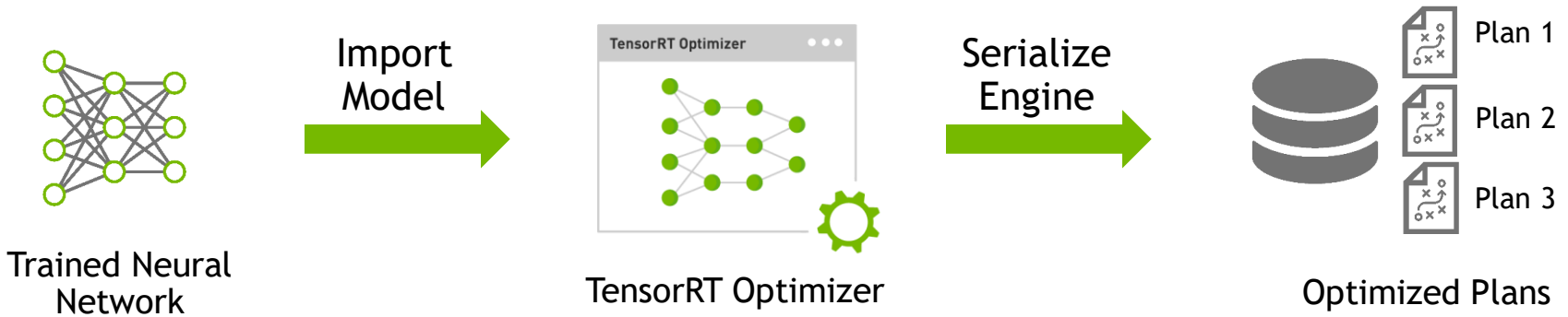
DEPLOYMENT

- Neural network training and inference

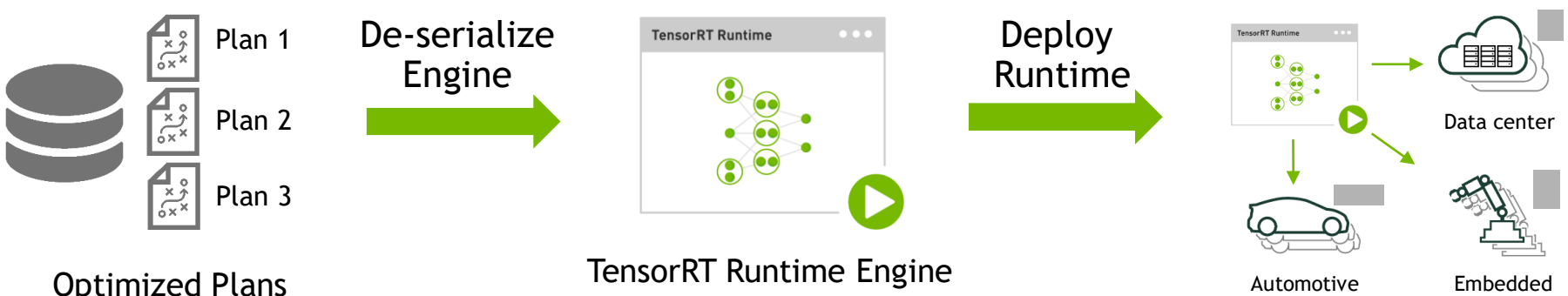


DEPLOYMENT

Step 1: Optimize trained model



Step 2: Deploy optimized plans with runtime



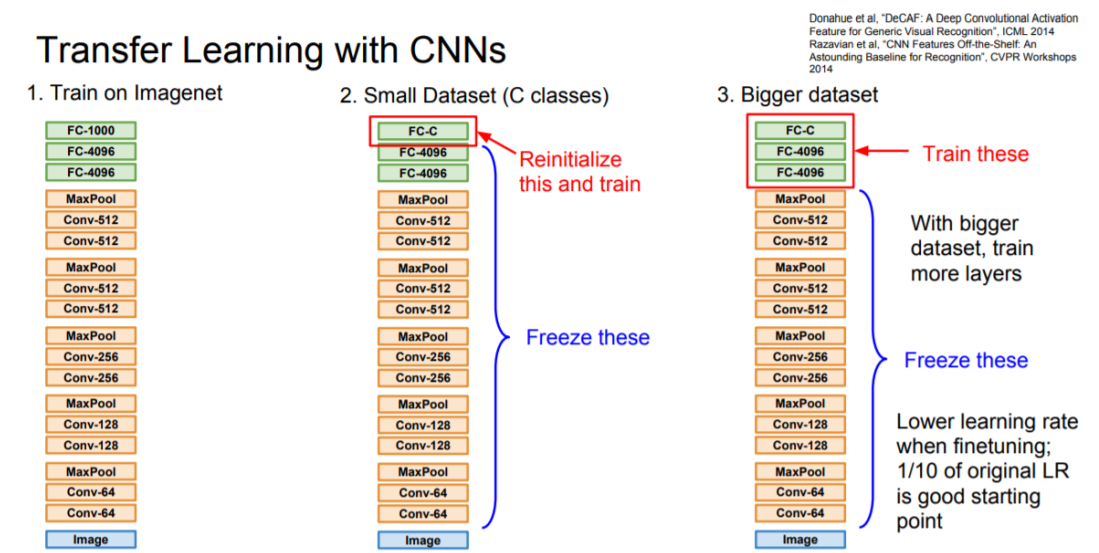
Task 3
DEPLOYMENT

TRANSFER LEARNING

TRANSFER LEARNING

- Why do we use **“transfer learning”**?
  → 충분한 양의 데이터를 갖고 있기가 쉽지 않음
  → 다른 사람들이 **미리 학습한 모델**을 이용

- Major types of transfer learning
  1. ConvNet as fixed feature extractor
  2. Fine-tuning the ConvNet
  3. Pretrained models



http://cs231n.github.io/transfer-learning/
http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture07.pdf

TRANSFER LEARNING

- Major types of transfer learning
  1. ConvNet as fixed feature extractor
     - Pretrained된 ConvNet의 마지막 Fully-connected layer 제거
     - 이 network를 new dataset의 feature extractor로 사용
     - new dataset에 대해 linear classifier 학습 (Linear SVM, Softmax classifier, …)
  2. Fine-tuning the ConvNet
  3. Pretrained models

http://cs231n.github.io/transfer-learning/

TRANSFER LEARNING

- Major types of transfer learning
  1. ConvNet as fixed feature extractor
  2. Fine-tuning the ConvNet
     - Classifier on top of the ConvNet을 new dataset에 대해 replace & retrain
     - Fine-tune the weights of the pretrained network by continuing the backpropagation
       - Fine-tune **all the layers** of the ConvNet
       - Keep some of **the earlier layers fixed**
       - Fine-tune **some higher-level portion** of the network
  3. Pretrained models

http://cs231n.github.io/transfer-learning/

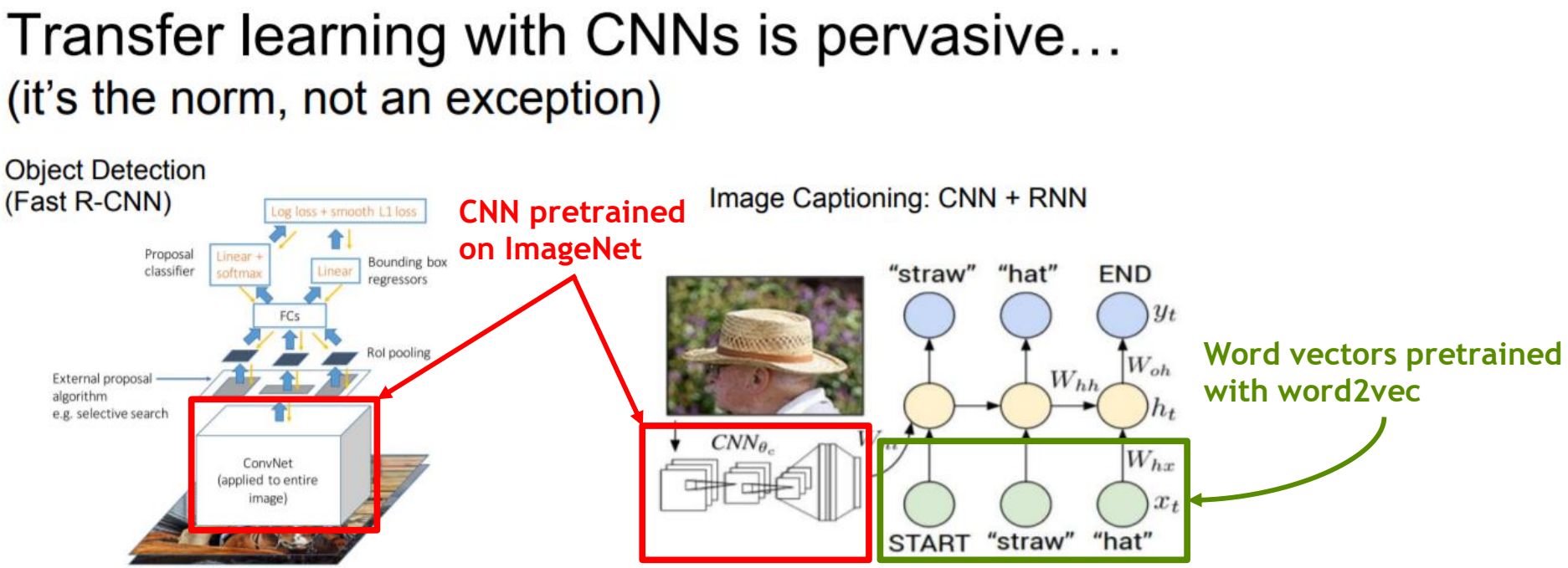
TRANSFER LEARNING

- Major types of transfer learning
  1. ConvNet as fixed feature extractor
  2. Fine-tuning the ConvNet
  3. Pretrained models
     - ImageNet으로 미리 학습한 모델 사용

http://cs231n.github.io/transfer-learning/

TRANSFER LEARNING

- Examples of transfer learning



http://cs231n.github.io/transfer-learning/

Reference

- http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture08.pdf
- http://cs231n.github.io/neural-networks-3/
- https://nittaku.tistory.com/289
- https://laonple.blog.me/220527647084
- https://laonple.blog.me/220542170499
- https://m.etnews.com/20171128000218
- https://www.samsungsds.com/global/ko/support/insights/1196843_2284.html
- https://becominghuman.ai/building-an-image-classifier-using-deep-learning-in-python-totally-from-a-beginners-perspective-be8dbaf22dd8
- http://cs231n.github.io/transfer-learning/
- http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture07.pdf

www.nvidia.com/dli