DRAG
Drone and Robotics Aziz Group

أكاديمية كاوست
KAUST ACADEMY

Practical Tools for Machine Learning

October 23, 2022

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Introduction

أكاديمية كاوست
KAUST ACADEMY

- Practical Implementation of Deep Learning algorithms is just as much an art as it is a science.
- The main take aways if not to start from scratch rather to build on top of the previous knowledge.
- Today, we will look at some important tools used in the practical implementation of Deep Learning algorithms.

Outline

- Data Handling
- Data Augmentation
- Transfer Learning
- Ensembling
- Batch Normalization

Data Handling

•As we have previously established that Deep Learning has been made possible by large amount of data and computational resourse

•An important aspect to keep in mind is the data handling:

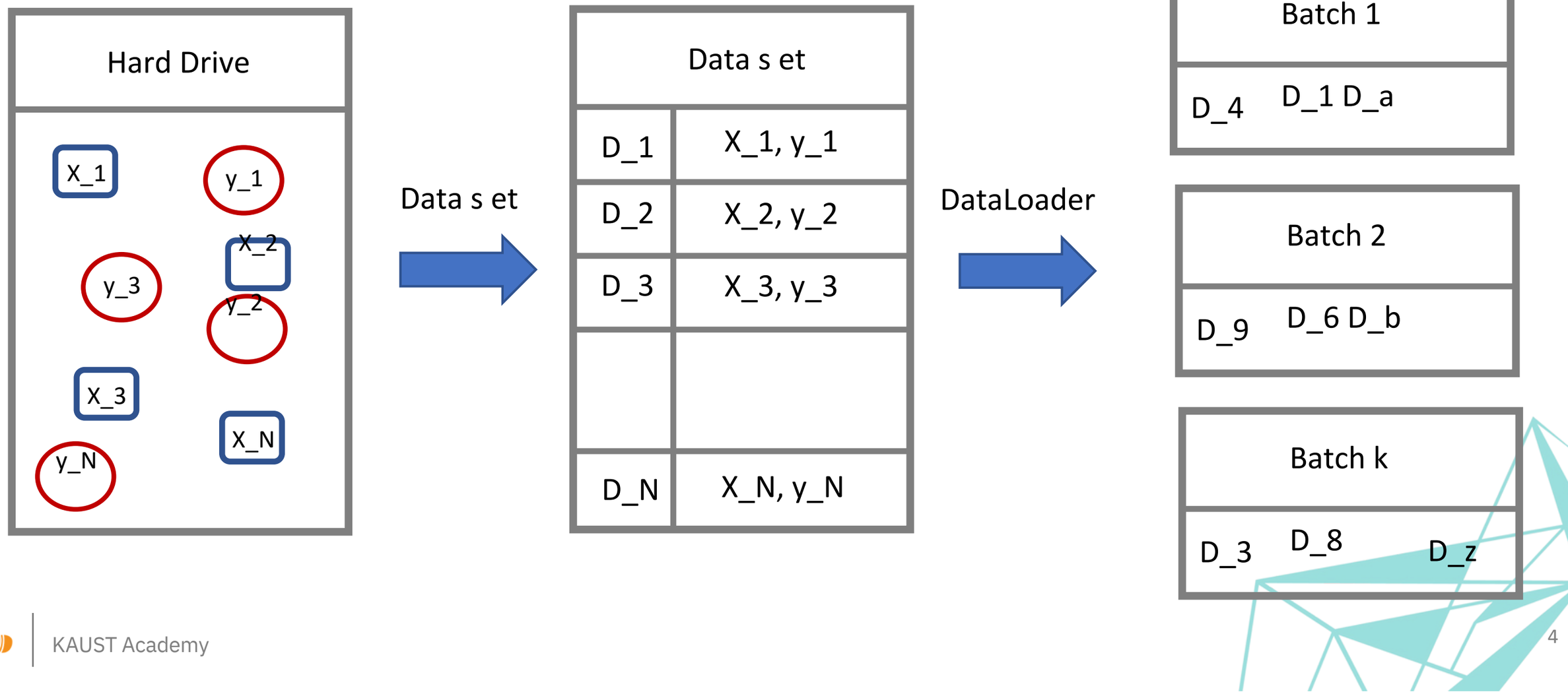
•How do we handle large amounts of data?

- •How to we read different components of data (from possible different parts of our hard drive) and provide it to our training algorithms?
- •How do we feed this data to SGD algorthims in a streamlined manner?

•PyTorch provides Dataset and DataLoaders to handle data in an efficient manner.

•We will extend the Dataset and DataLoaders class to construct our own Dataloaders

DataLoaders



Data Augmentation

أكاديمية كاوست
KAUST ACADEMY

- Data is the fundamental building block of any machine learning algorithm
- In several applications we don't have access to unlimited data
- So we use Data Augmentation techniques to improve the preformance of our models
- Note: It is better to spend time on data rather than fine-scale architecture search in deep learning

Data Augmentation

- Create *virtual* training samples
- Horizontal flip
- Random crop
- Color casting
- Geometric distortion
- Translation
- Rotation

① Efficient Training of Deep Networks

① problems :- Overfitting

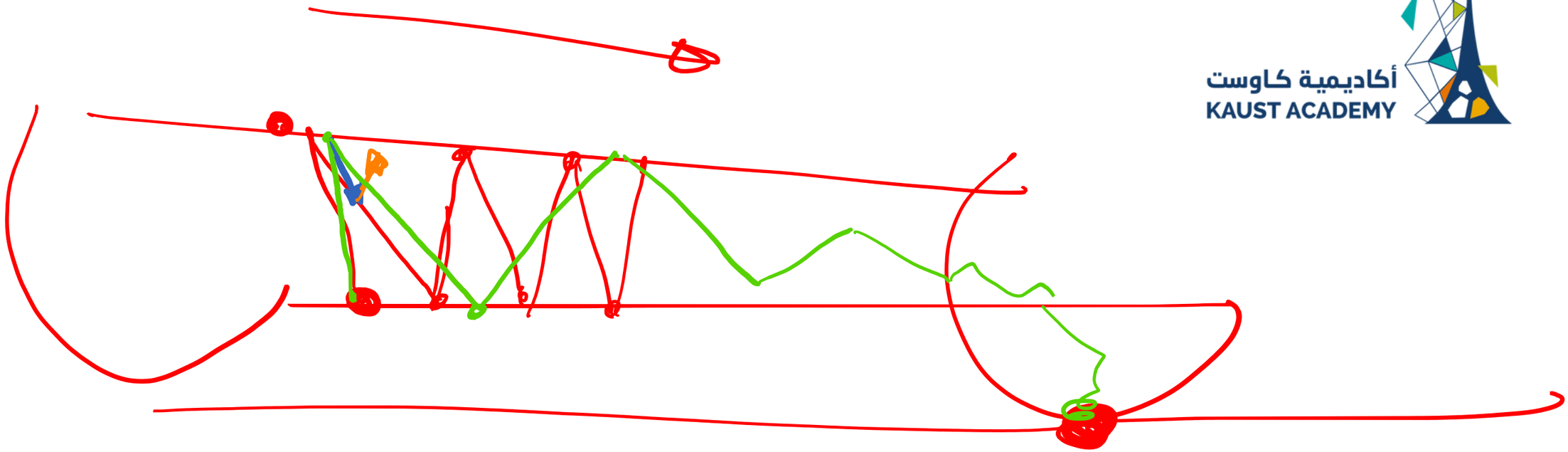
② Gradient → too large

→ too small

③ learning rate

① learning rate schedules

② Momentum



$$v^0 = -\eta g^0$$

$$v^1 = (1-m)(-\eta g^1) + m(v^0)$$

$$v^2 = (1-m)(-\eta g^2) + m(v^1)$$

$$v^0 = -\eta g^0$$

$$v^1 = (1-m)(-\eta g^1) + m v^0$$

$$v^2 = (1-m)(-\eta g^2) + m v^1$$

$$v^0 = -\eta g^0$$

$$v^1 = \frac{(1-m)(-\eta g^1)}{+m(-\eta g^0)}$$

$$v^2 = (1-m)(-\eta g^2) + m(1-m)(-\eta g^1) + m(m(-\eta g^0)$$

$$v^2 = (1-m)(-\eta g^2) + m(1-m)(-\eta g^1) + m^2(-\eta g^0)$$

① Regularization

$w^2 = 1 - \lambda k$

$w' = k + 1$

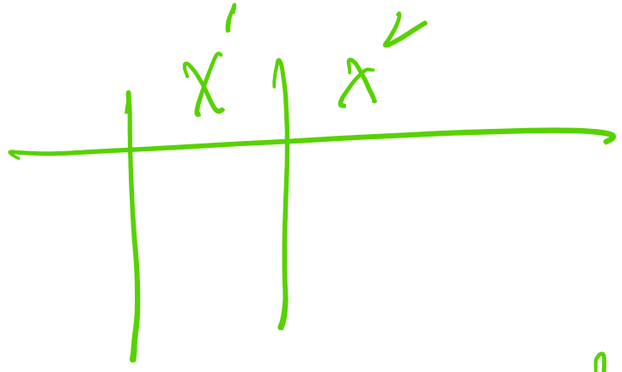
ou $w'^{(2)} + w^{2(2)}$

$$\hat{y} = w^1 x^1 + w^2 x^2$$

$$y = x^1 + x^2$$

$\lambda \in [0, 2]$

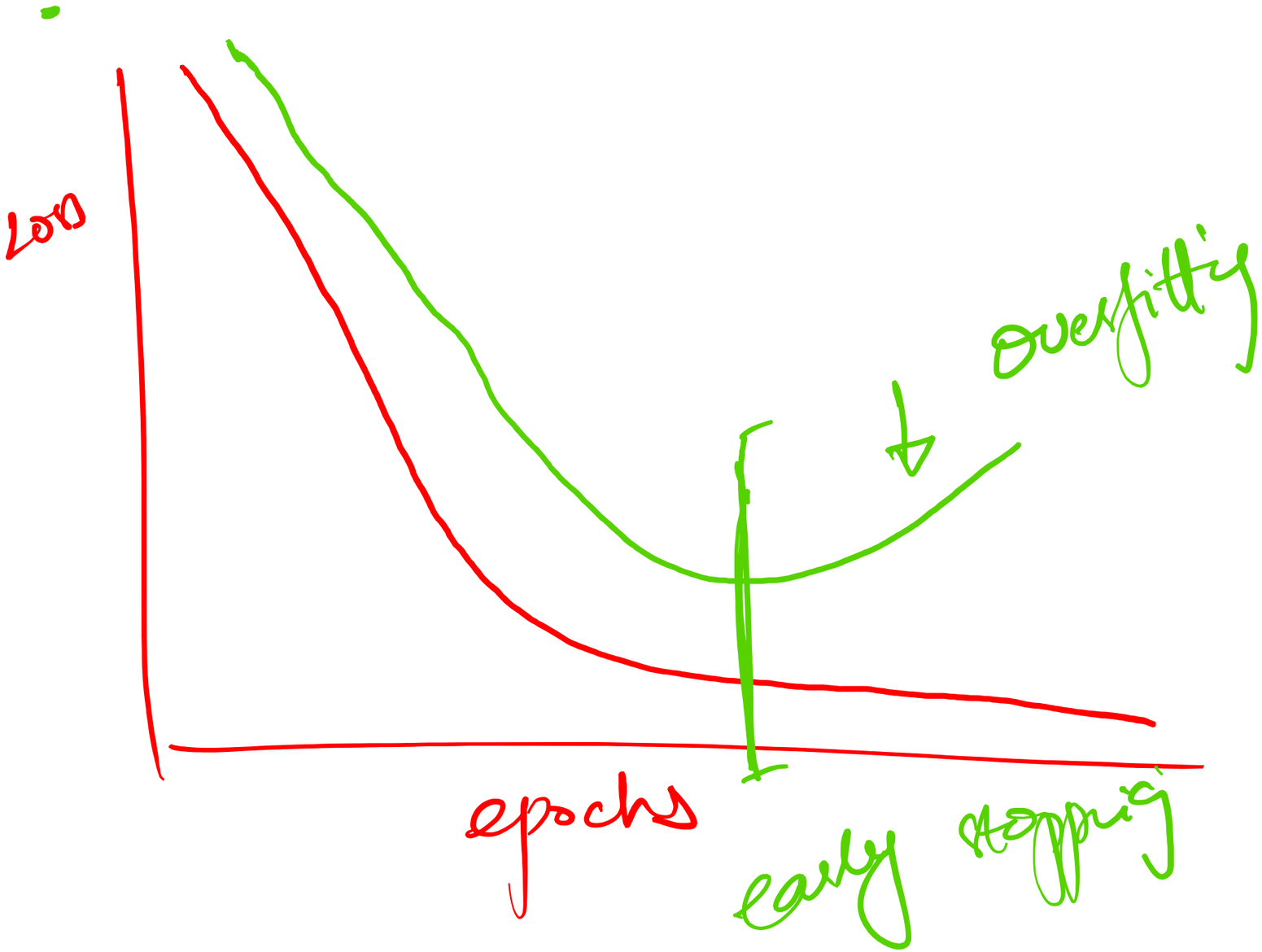
$w^1 = 1$	$w^2 = 1$

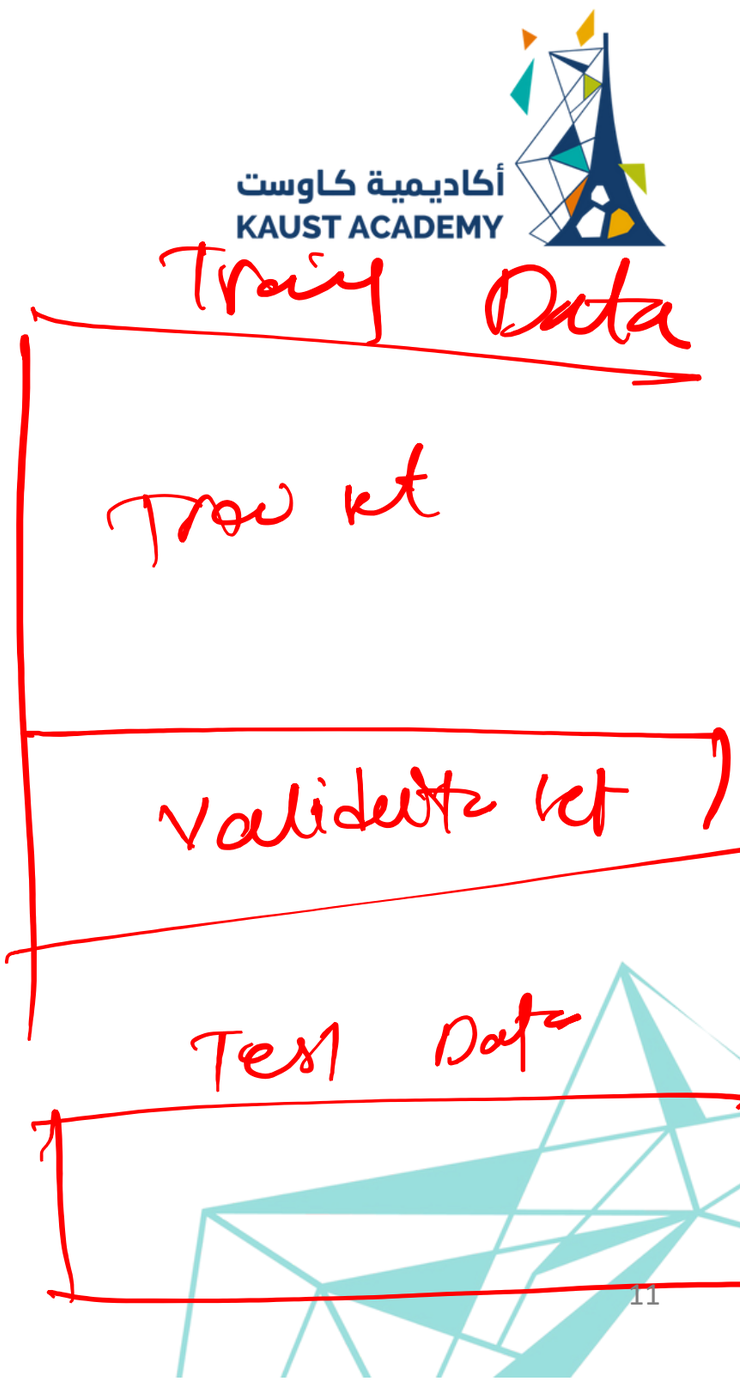


$w^1 = 2$ $\quad w^2 = 1 - 1$

$x^2 = 2x^1$

$$2x^1 + x^2 - 2x^2 = x^1 + x^2$$


Loss
overfitting
epochs
early stopping


Training Data
Trained
Validation set
Test Data
prohibited

1. Dropout
2. Batch Normalization

2%

أكاديمية كاوست
KAUST ACADEMY

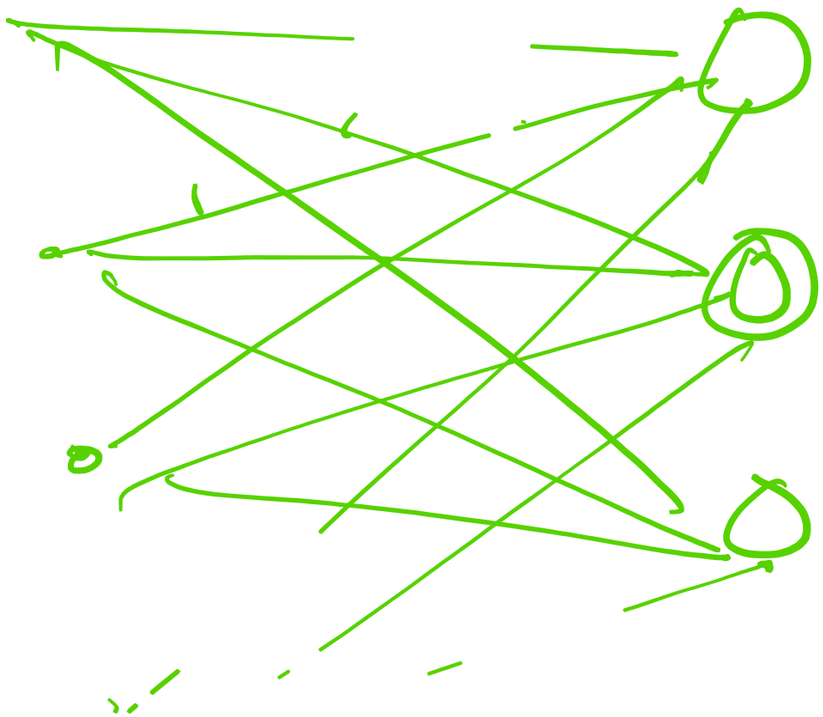
Ensemble

DNN 1 | DNN 2 | DNN 3

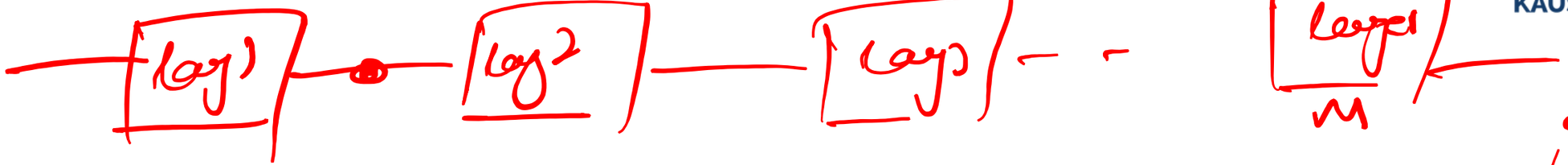
$(0.02)(0.02)(0.02) + \binom{3}{2} 0.98 (0.02)^2$

$8 \times 10^{-6} + 3 \times 6 \times 10^{-4}$

$0.000208 \quad 12 \times 10^{-4} + 8 \times 10^{-6}$



$x \sim N(0,1)$



$a^1 \not\sim N(0,1)$ $\quad \tilde{a}^1 = \dfrac{a^1 - \mu}{\sigma}$ $\quad \lambda \tilde{a} + b$

a^1

$\mu = \text{mean}(a^1)$

$\sigma^2 = \text{var}(a^1)$

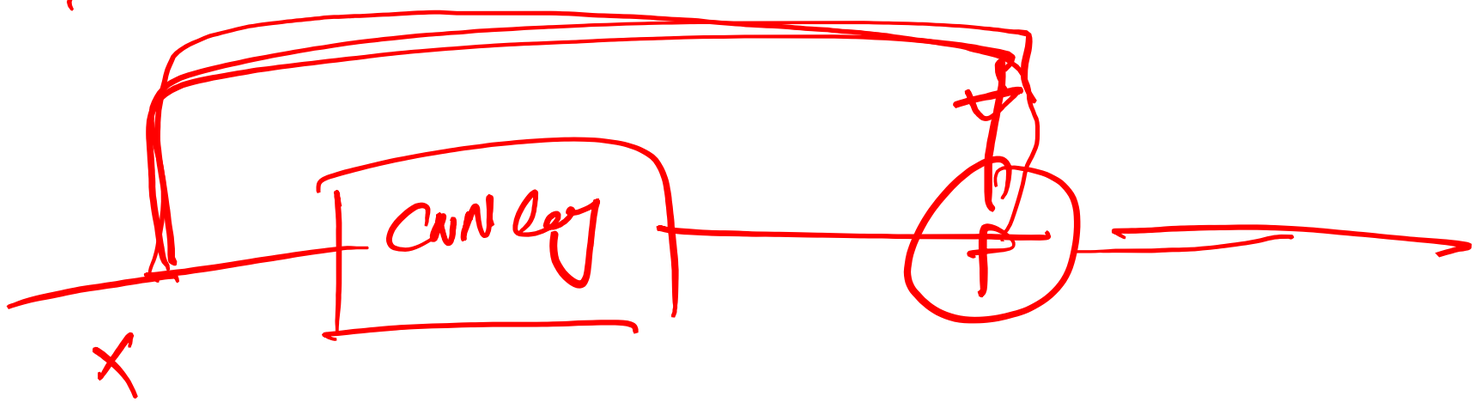
Transfer learning (fine tuning)

1950's

① (Leak - Relu) (Relu)

+ Alex Net, VGG, Google inception

② Res Net



Transfer Learning

•Improvement of learning in a newtask through the *transfer of knowledge* from a related task that has already been learned.

•We will look at one strategy of transfer learning called Fine-Tuning

When to fine-tune your model?

•New dataset is small with distribution similar to original dataset.

•Keep the feature extraction part fixed and fine-tune the classifier part of the network

•New dataset is large with similar distribution to the original dataset

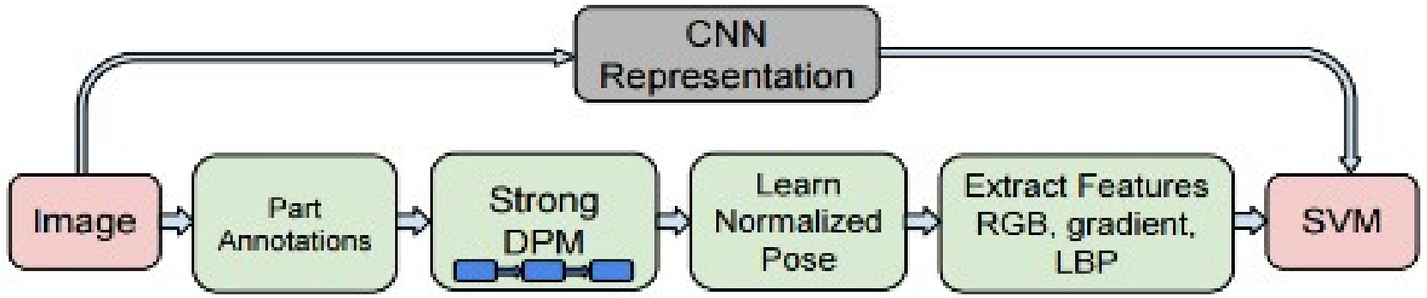
•Fine tune both the feature extractor and the classifier part of the network

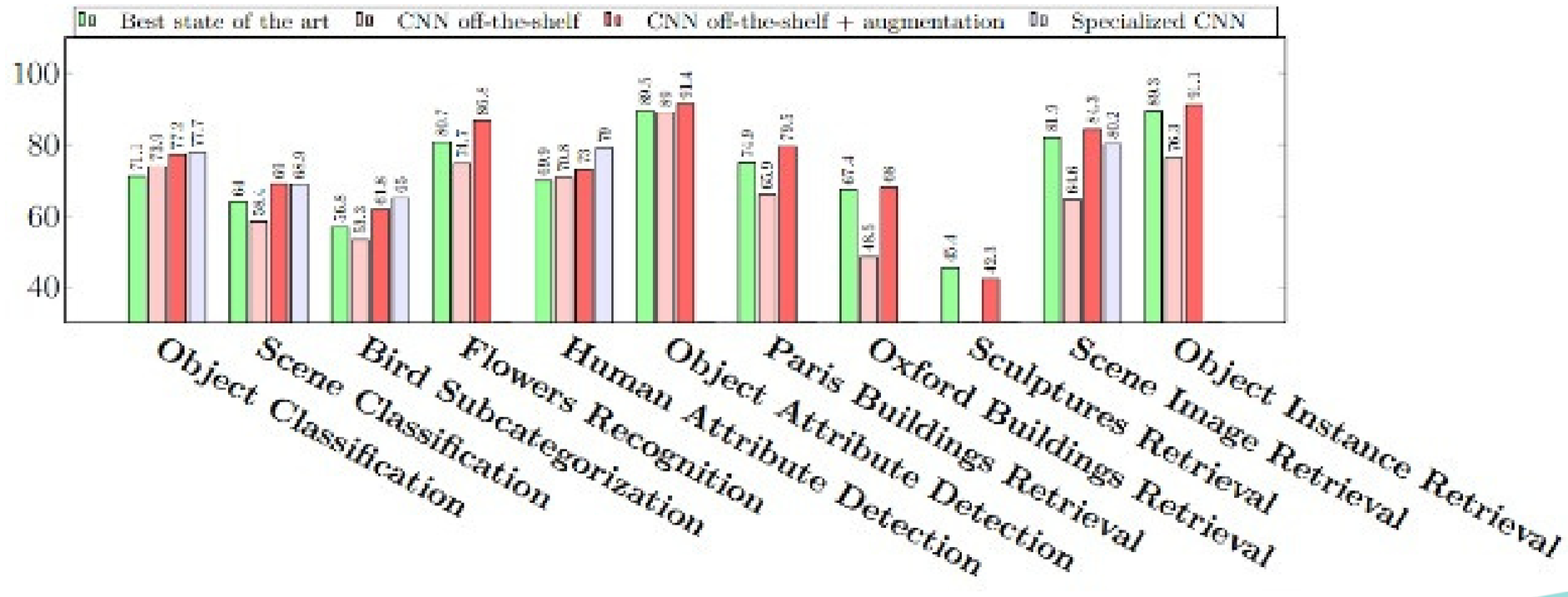
•New dataset is small but different distribution from the original dataset

•Use SVM classifier on the features extracted from the feature extractor part of the Network

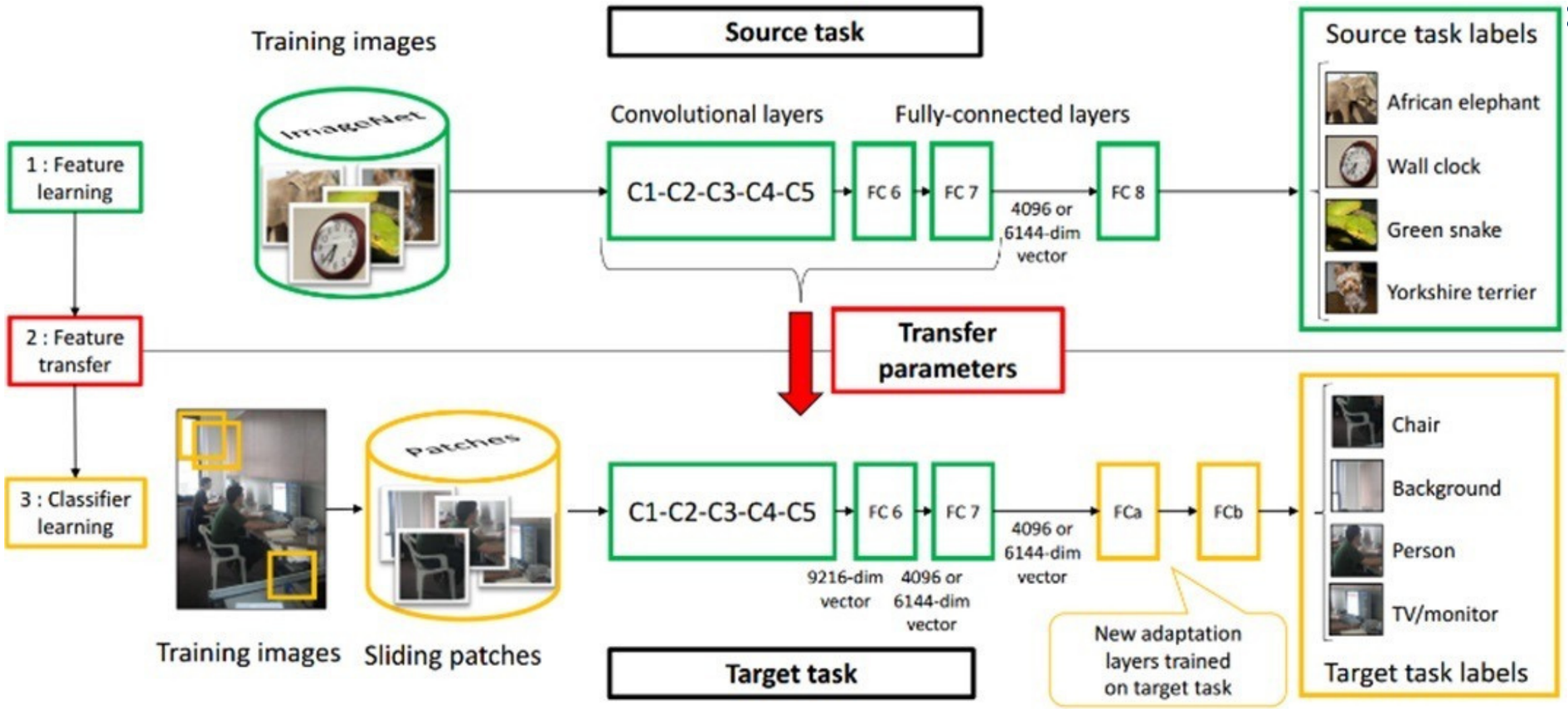
•New dataset is large and different distribution from the original dataset

•Fine tune both the feature extractor and the classifier part of the network





CNN Features off-the-shelf: an Astounding Baseline for Recognition [Razavianet al. 2014]



Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks [Oquab et al. CVPR 2014]

Ensembling

- Team-work is the best policy
- Multiple networks for the same task
- Max Voting for final classification

Ensembling: A simple analysis

•Let’s assume that we have a test dataset with N elements and an ensemble of M models.

•Also assume that the probability of error of the label for an image on a model in the ensemble is dentoted by p(e) and is i.i.d

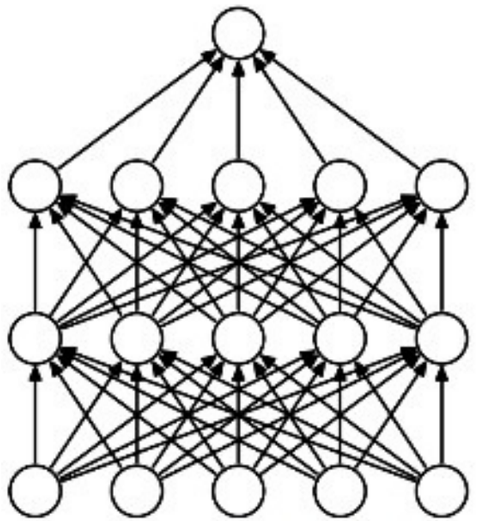
•For an example assume M=3 and e =0.01

•Then probability of error of label for the max voting ensemble will be

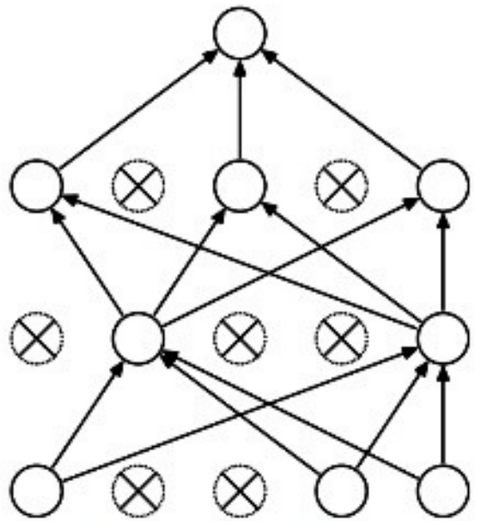
- For the above example p(e)=0.0003, which is significantly lower than a single model

$$p(e) = 1 \quad (1 \quad e)^3 \quad \sqrt{3}_2 \blacklozenge (1 \quad e)^2 e$$

Dropout



(a) Standard Neural Net



(b) After applying dropout.

Intuition: successful conspiracies

- 50 people planning a conspiracy

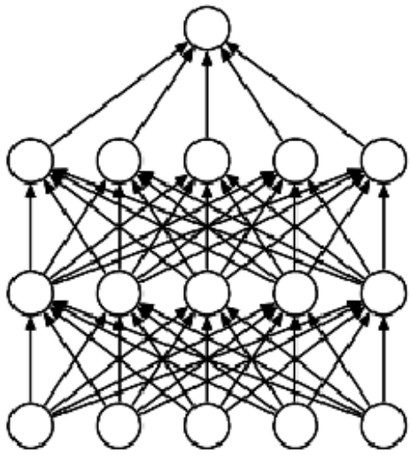
- Strategy A: plan a big conspiracy involving 50 people
  - Likely to fail. 50 people need to play their parts correctly.

- Strategy B: plan 10 conspiracies each involving 5 people

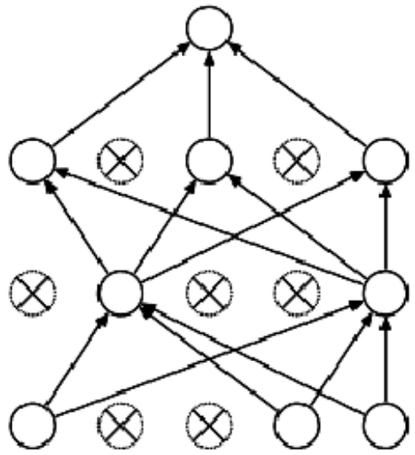
Likely to succeed!

Dropout: A simple way to prevent neural networks from overfitting[Srivastava JMLR 2014]

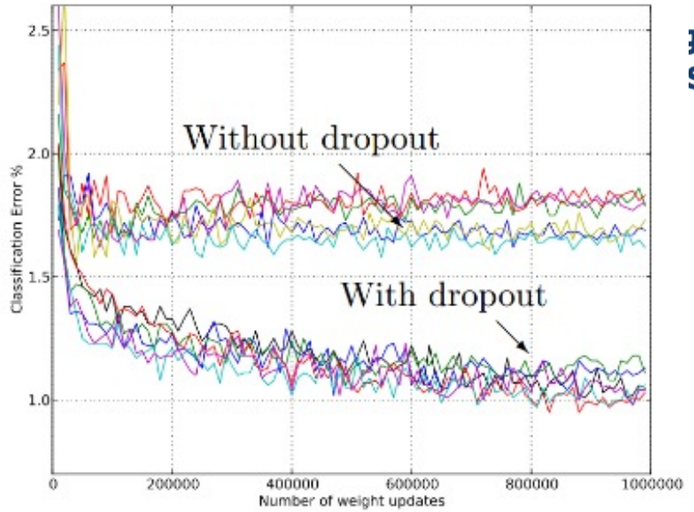
Dropout

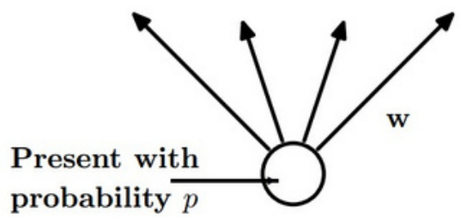


(a) Standard Neural Net

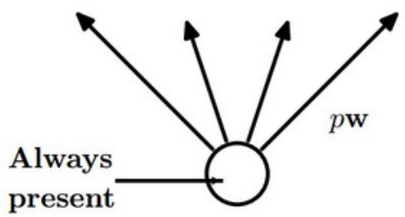


(b) After applying dropout.





(a) At training time



(b) At test time

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SVM on Fisher Vectors of Dense SIFT and Color Statistics	-	-	27.3
Avg of classifiers over FVs of SIFT, LBP, GIST and CSIFT	-	-	26.2
Conv Net + dropout (Krizhevsky et al., 2012)	40.7	18.2	-
Avg of 5 Conv Nets + dropout (Krizhevsky et al., 2012)	38.1	16.4	16.4

Table 6: Results on the ILSVRC-2012 validation/test set.

Main Idea: approximately combining exponentially many different neural network architectures efficiently

Dropout: A simple way to prevent neural networks from overfitting[Srivastava JMLR 2014]

Batch Normalization

أكاديمية كاوست
KAUST ACADEMY

- Very useful for training deep networks
- Standardizes input of each layer for a mini-batch
- Results in speed up of training
- For more detail, refer to the paper: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 2015

Batch Normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

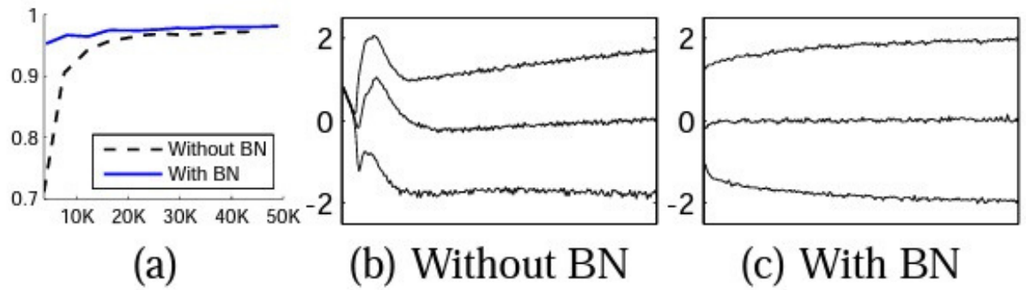
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$



(a) (b) Without BN (c) With BN

Figure 1: (a) *The test accuracy of the MNIST network trained with and without Batch Normalization, vs. the number of training steps. Batch Normalization helps the network train faster and achieve higher accuracy.* (b, c) *The evolution of input distributions to a typical sigmoid, over the course of training, shown as* $\{15, 50, 85\}$ *th percentiles. Batch Normalization makes the distribution more stable and reduces the internal covariate shift.*

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [Ioffe and Szegedy 2015]

Things to remember

- Training Deep Networks Dropout
  - •Data augmentation
  - •Activation
  - •Batch normalization
- Transfer learning
  - •Use Fine-tuning when possible

Full Deep learning Pipeline

•Data Pre-processing •
Architecture
• Loss
• Optimizer
• DataLoaders
•Data Augmentation
• Fine-Tuning
• Ensembling

Course Feedback

https://forms.gle/sQMCFUCgMVcJmAbp9