MACHINE LEARNING FOR SIGNAL PROCESSING

5-3-2025

Sriram Ganapathy
LEAP lab, Electrical Engineering, Indian Institute of Science,
sriramg@iisc.ac.in

....

Viveka Salinamakki, Varada R.
LEAP lab, Electrical Engineering, Indian Institute of Science

....

http://leap.ee.iisc.ac.in/sriram/teaching/MLSP25/

STORY SO FAR

EM algorithm

Classification Problem

Decision Theory

Generative Modeling

Gaussian Modeling

Gaussian Mixture Modeling

Function Modeling

Linear Models for Regression and Classification

Kernel Machines & Max-margin classifiers

Support Vector Machines

Data Representations PCA, LDA

Discriminative Modeling

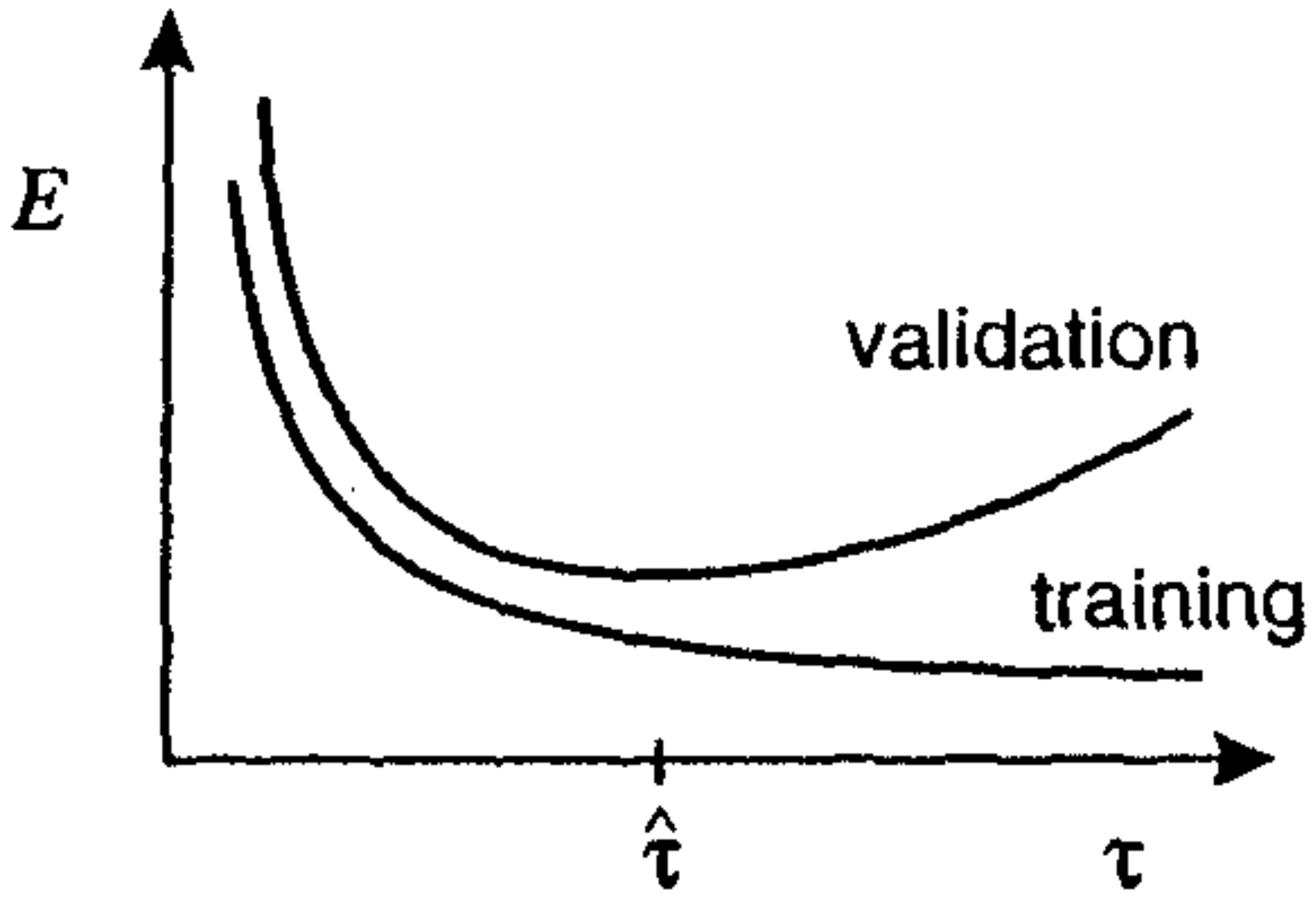
Gradient Descent

Neural Networks

Learning Rules & Normalization

Deep neural architectures

Regularization in Neural Networks



Other Approaches

- Training with noise
- Mixture of models
- Mixture of experts approach
- Dropout
- Learning rules

An overview of gradient descent optimization algorithms*

Sebastian Ruder
Insight Centre for Data Analytics, NUI Galway
Aylien Ltd., Dublin
ruder.sebastian@gmail.com

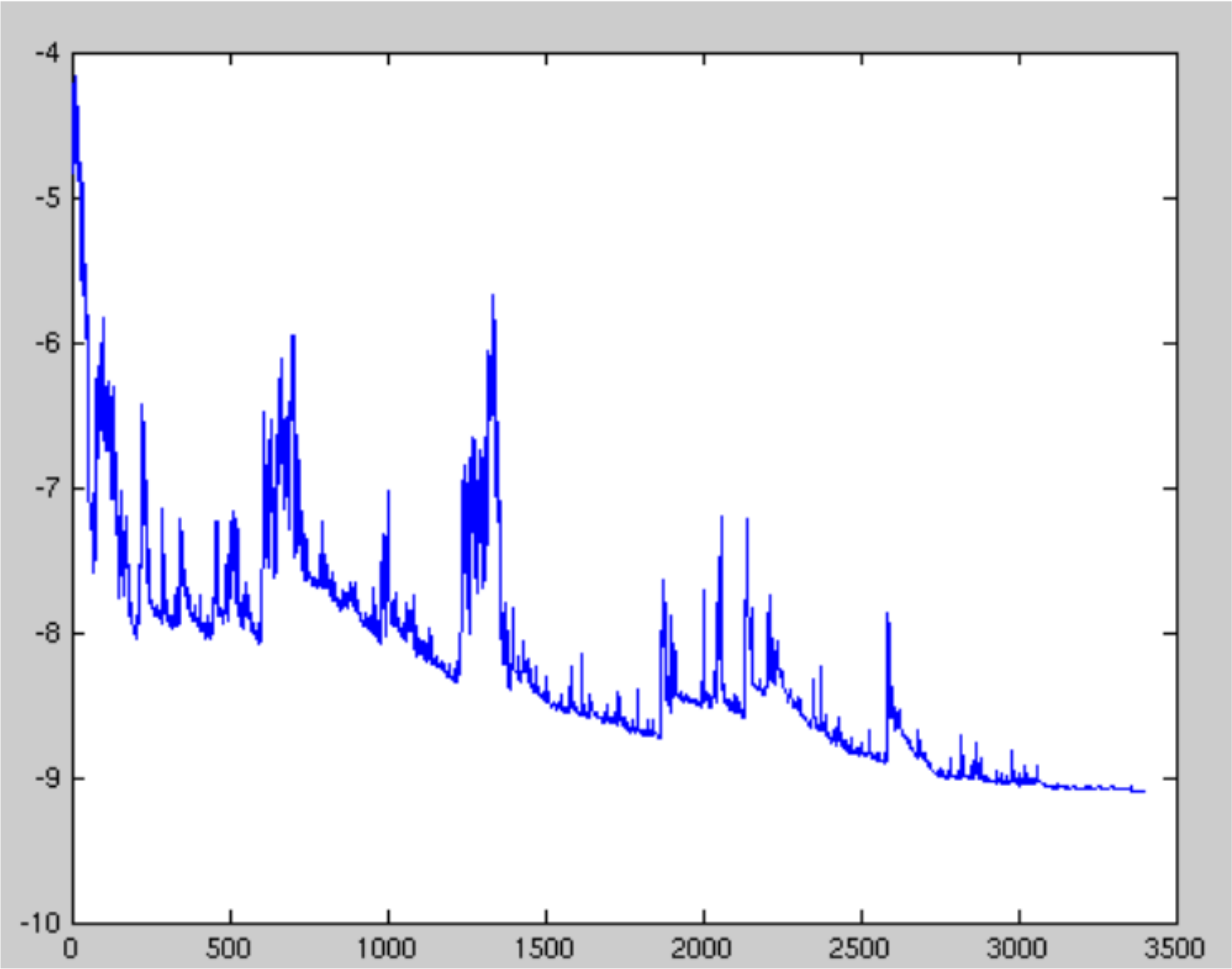
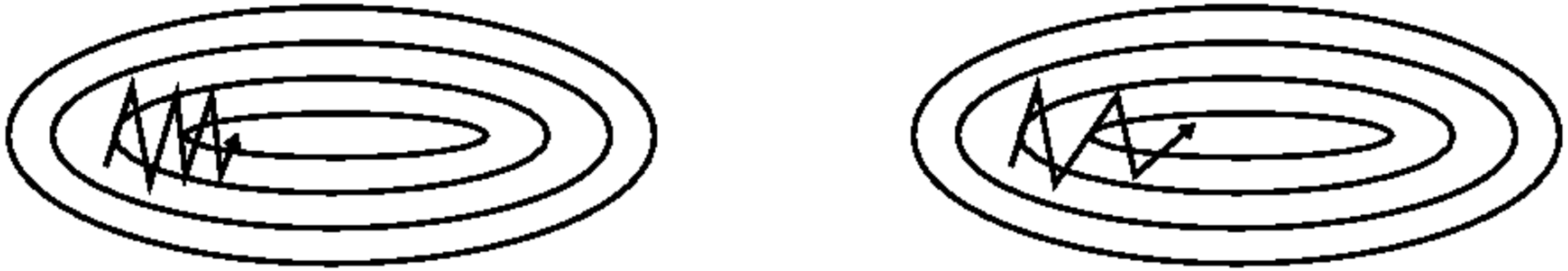


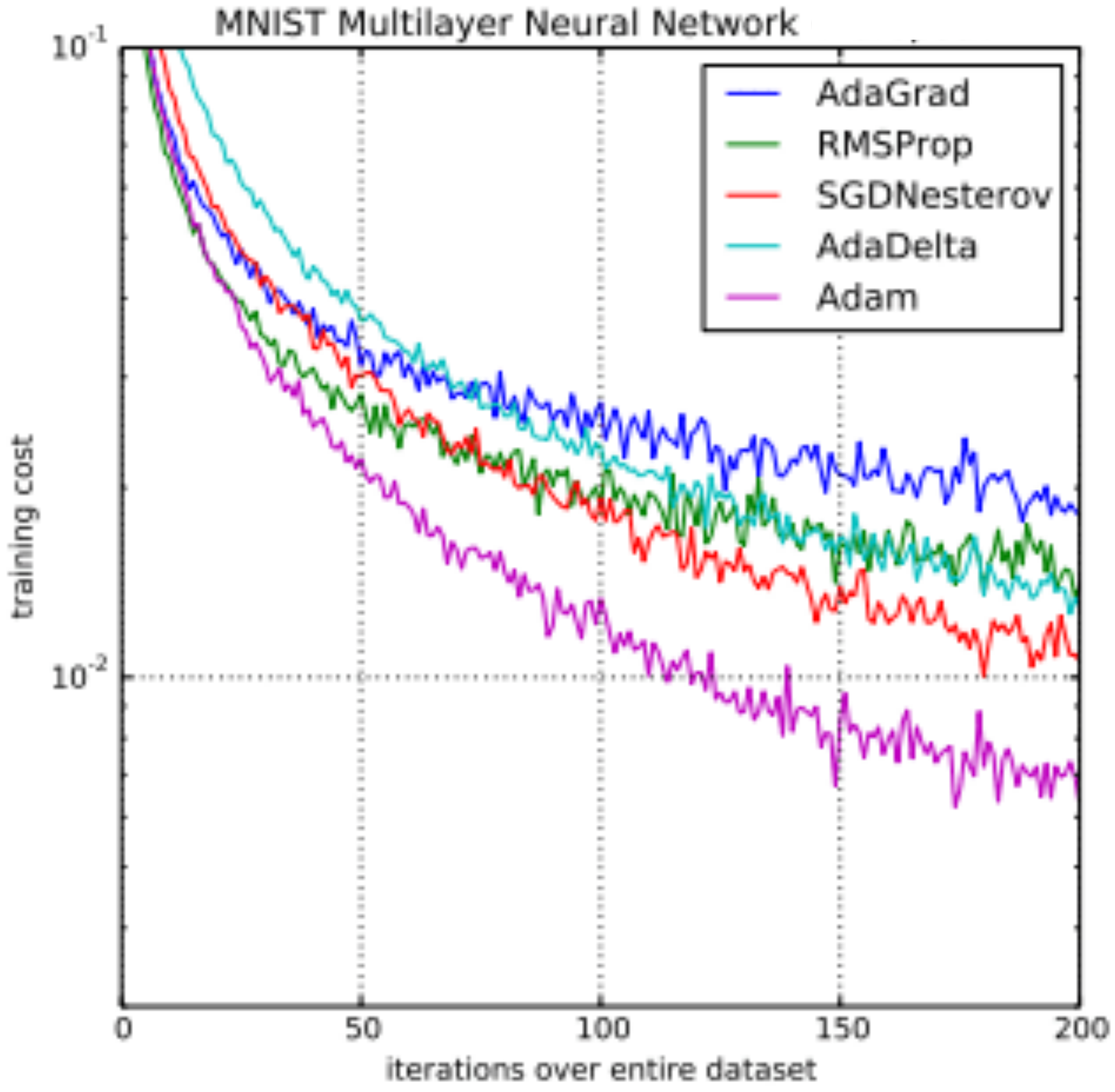
Figure 1: SGD fluctuation (Source: Wikipedia)

Momentum



(a) SGD without momentum (b) SGD with momentum

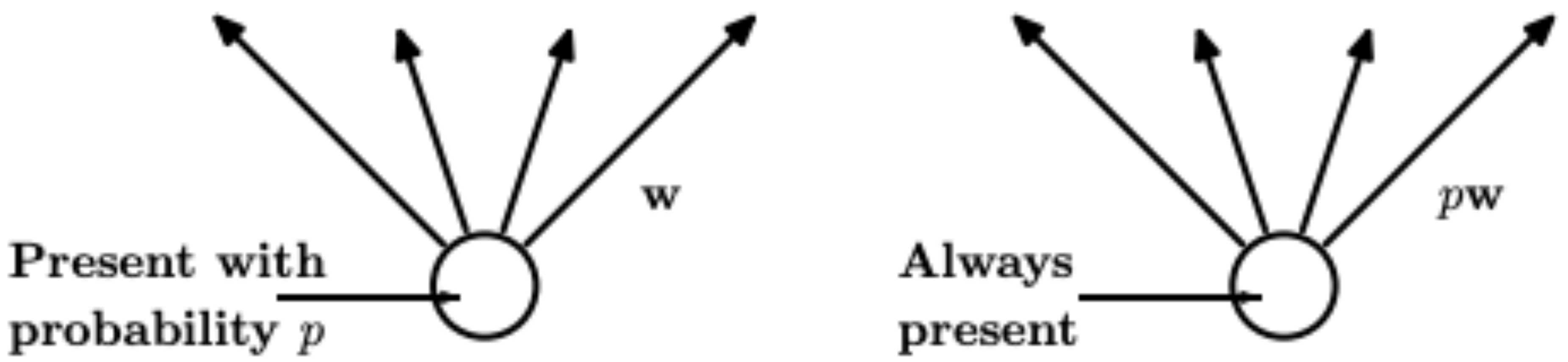
COMPARING DIFFERENT LEARNING RULES



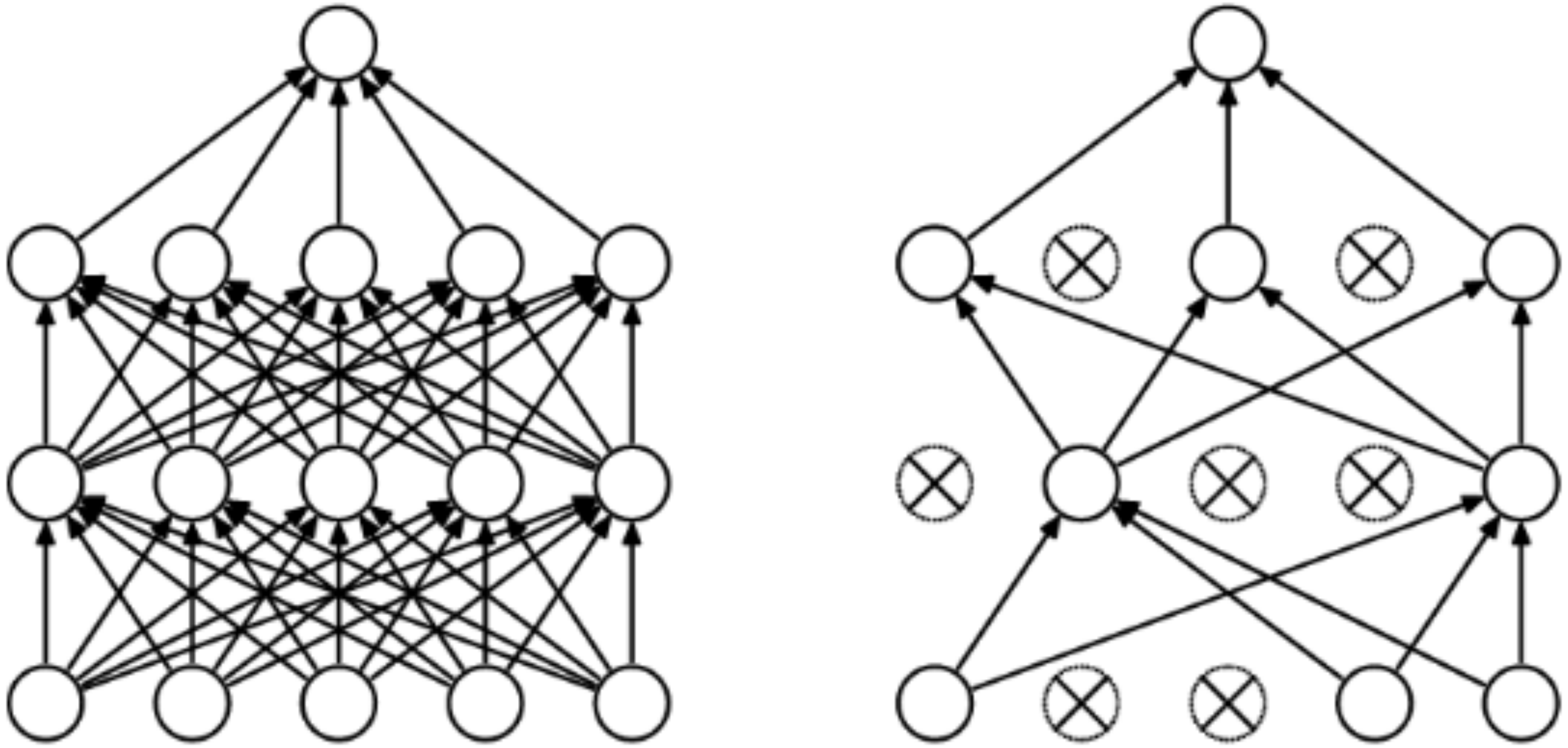
DROPOUT IN NEURAL NETWORKS

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Nitish Srivastava NITISH@CS.TORONTO.EDU
Geoffrey Hinton HINTON@CS.TORONTO.EDU
Alex Krizhevsky KRIZ@CS.TORONTO.EDU
Ilya Sutskever ILYA@CS.TORONTO.EDU
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
10 Kings College Road, Rm 3302
Toronto, Ontario, M5S 3G4, Canada.



(a) At training time (b) At test time

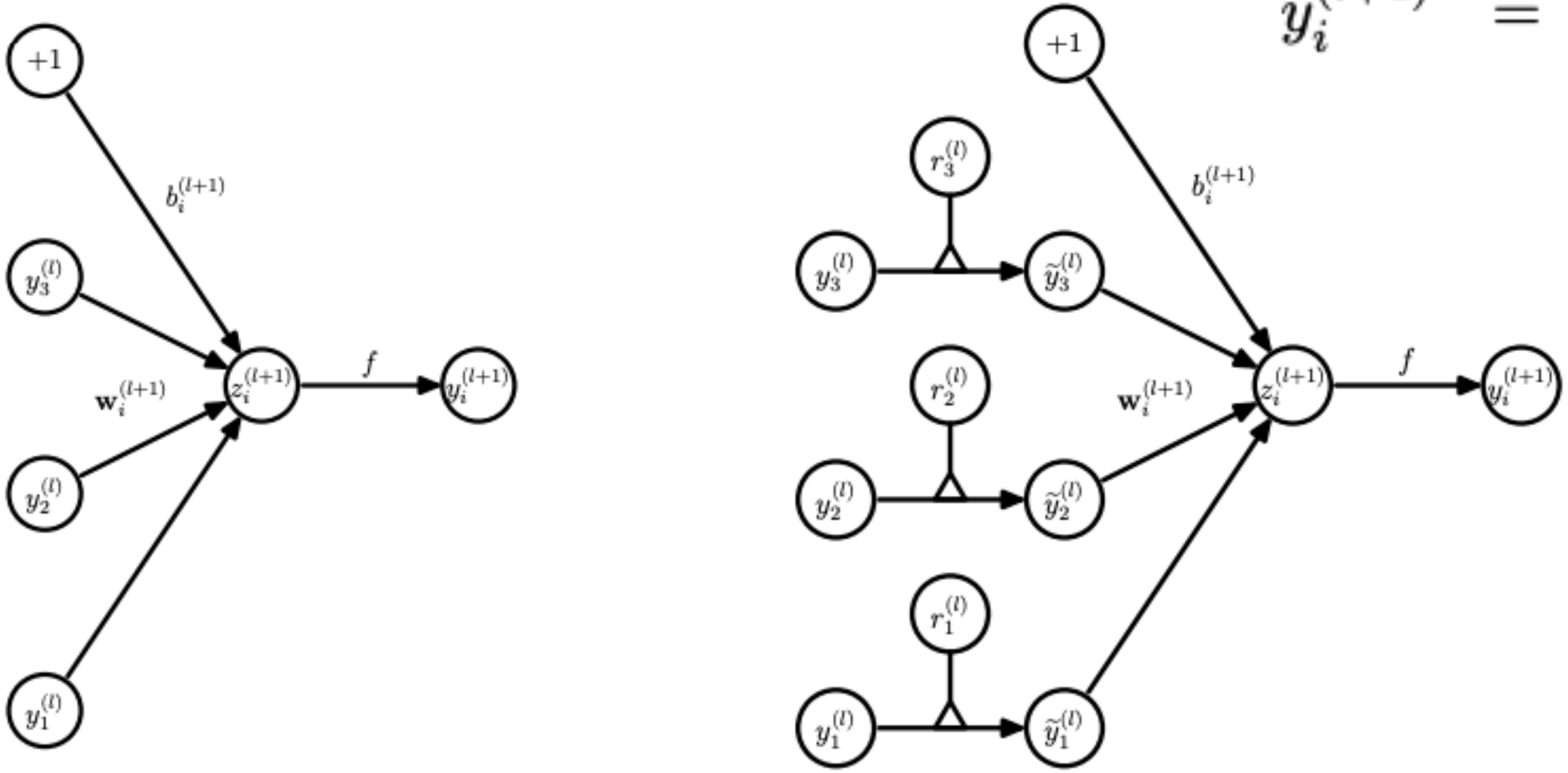


(a) Standard Neural Net (b) After applying dropout.

STANDARD VS DROPOUT NETWORKS

$$z_i^{(l+1)} = \mathbf{w}_i^{(l+1)}\mathbf{y}^l + b_i^{(l+1)},$$
$$y_i^{(l+1)} = f(z_i^{(l+1)}),$$

$$r_j^{(l)} \sim \text{Bernoulli}(p),$$
$$\widetilde{\mathbf{y}}^{(l)} = \mathbf{r}^{(l)} * \mathbf{y}^{(l)},$$
$$z_i^{(l+1)} = \mathbf{w}_i^{(l+1)}\widetilde{\mathbf{y}}^l + b_i^{(l+1)},$$
$$y_i^{(l+1)} = f(z_i^{(l+1)}).$$



(a) Standard network

(b) Dropout network

DROPOUT IN NEURAL NETWORKS

Random nodes are removed from the forward computation at dropout rate

STANDARD VERSUS DROPOUT

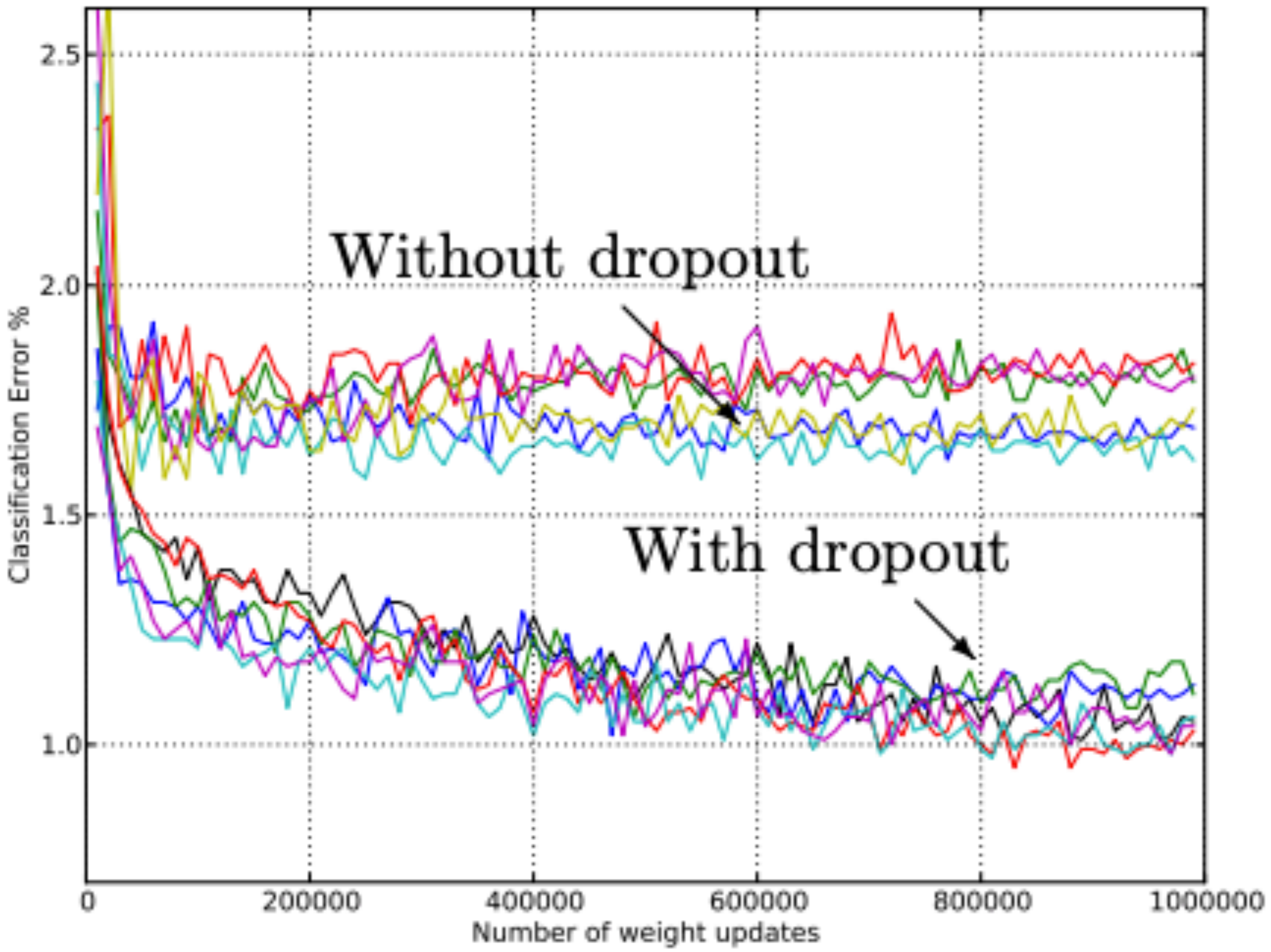


Figure 4: Test error for different architectures with and without dropout. The networks have 2 to 4 hidden layers each with 1024 to 2048 units.

NORMALIZATION TECHNIQUES

Batch Normalization and Layer Normalization

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

Sergey Ioffe
Google Inc., *sioffe@google.com*

Christian Szegedy
Google Inc., *szegedy@google.com*

Batch Normalization

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

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

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

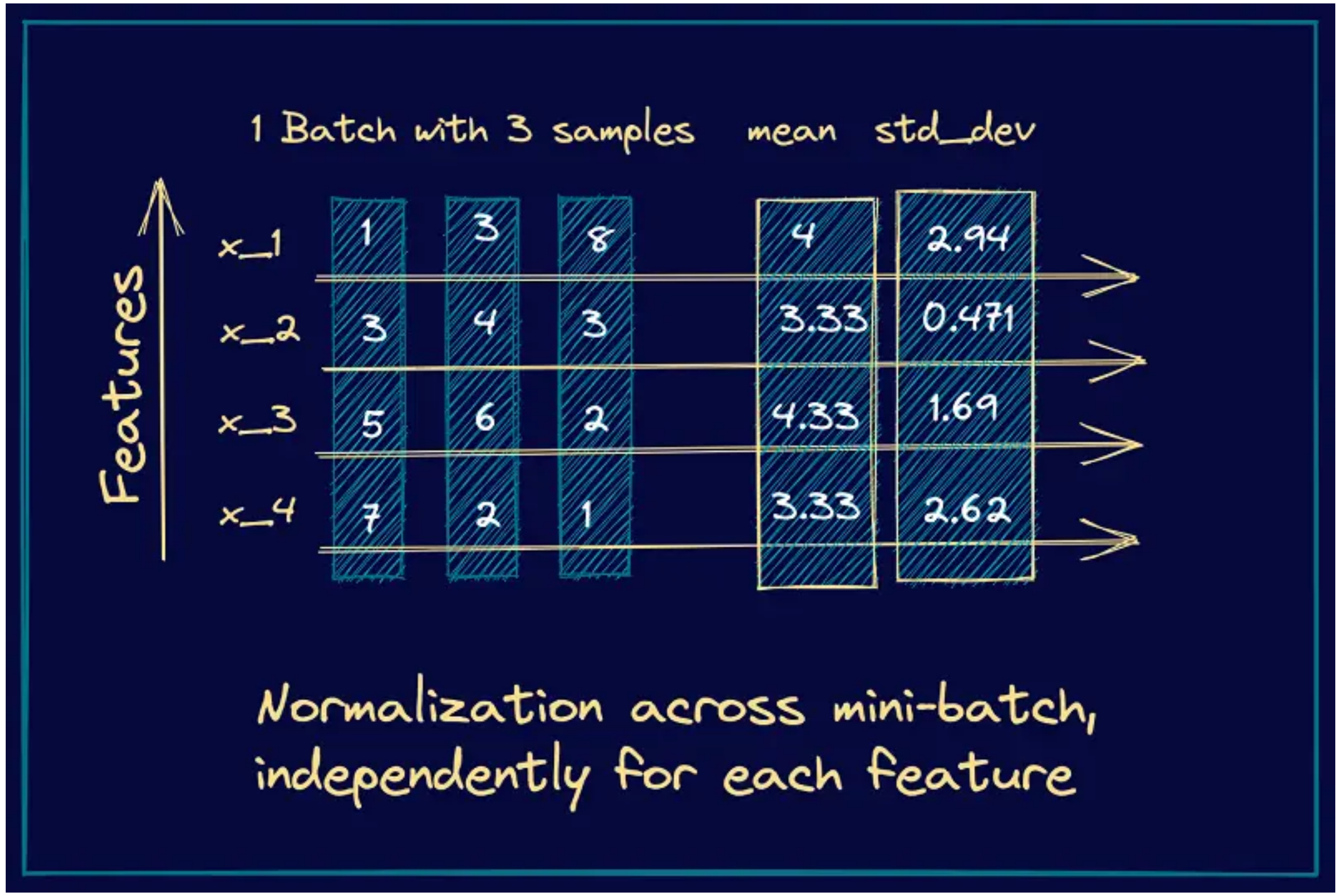
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

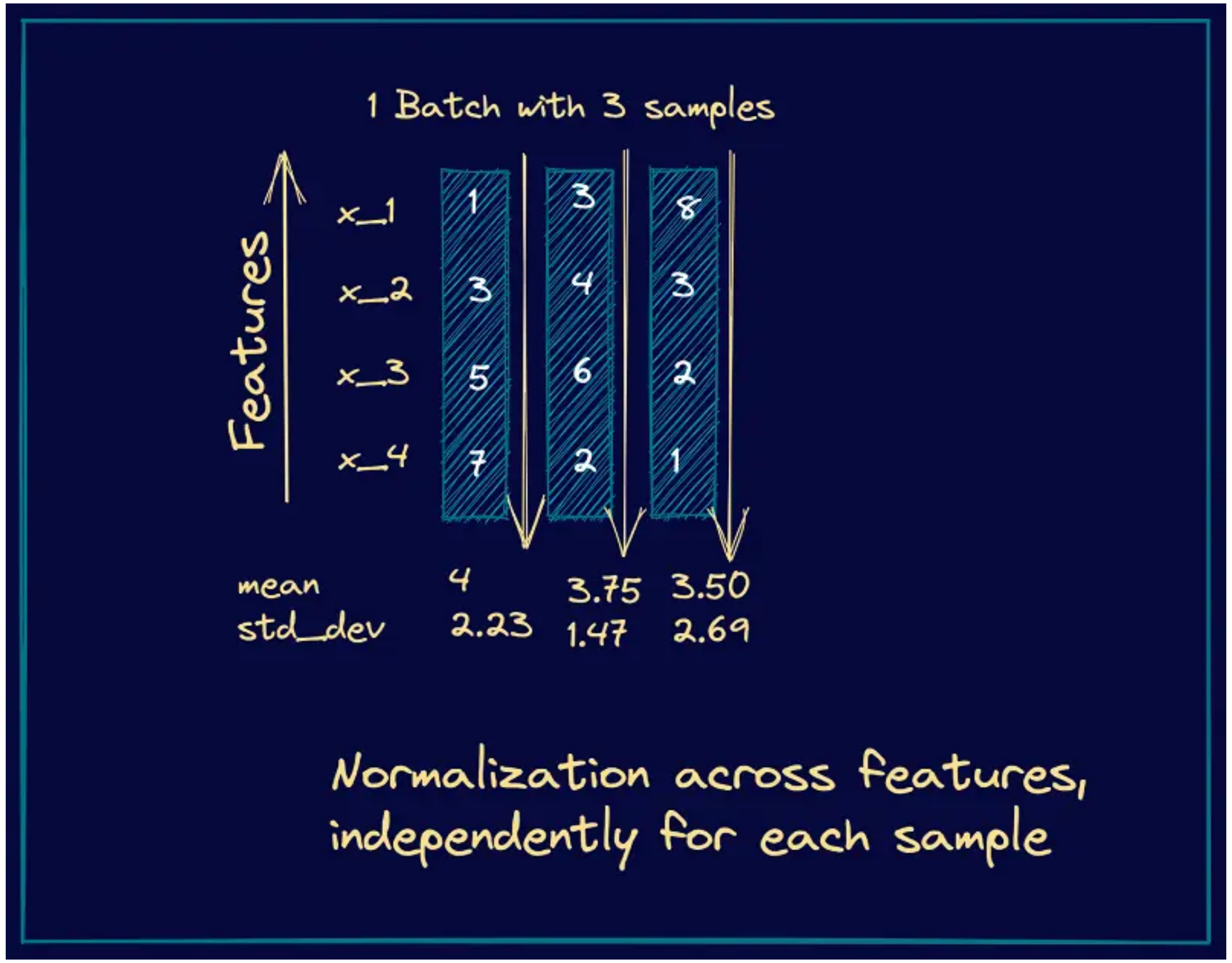
$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

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

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

COMPARING DIFFERENT NORMALIZATION





https://www.pinecone.io/learn/batch-layer-normalization/

COMPARING DIFFERENT NORMALIZATION

- Batch normalization normalizes each feature independently across the mini-batch. Layer normalization normalizes each of the inputs in the batch independently across all features.

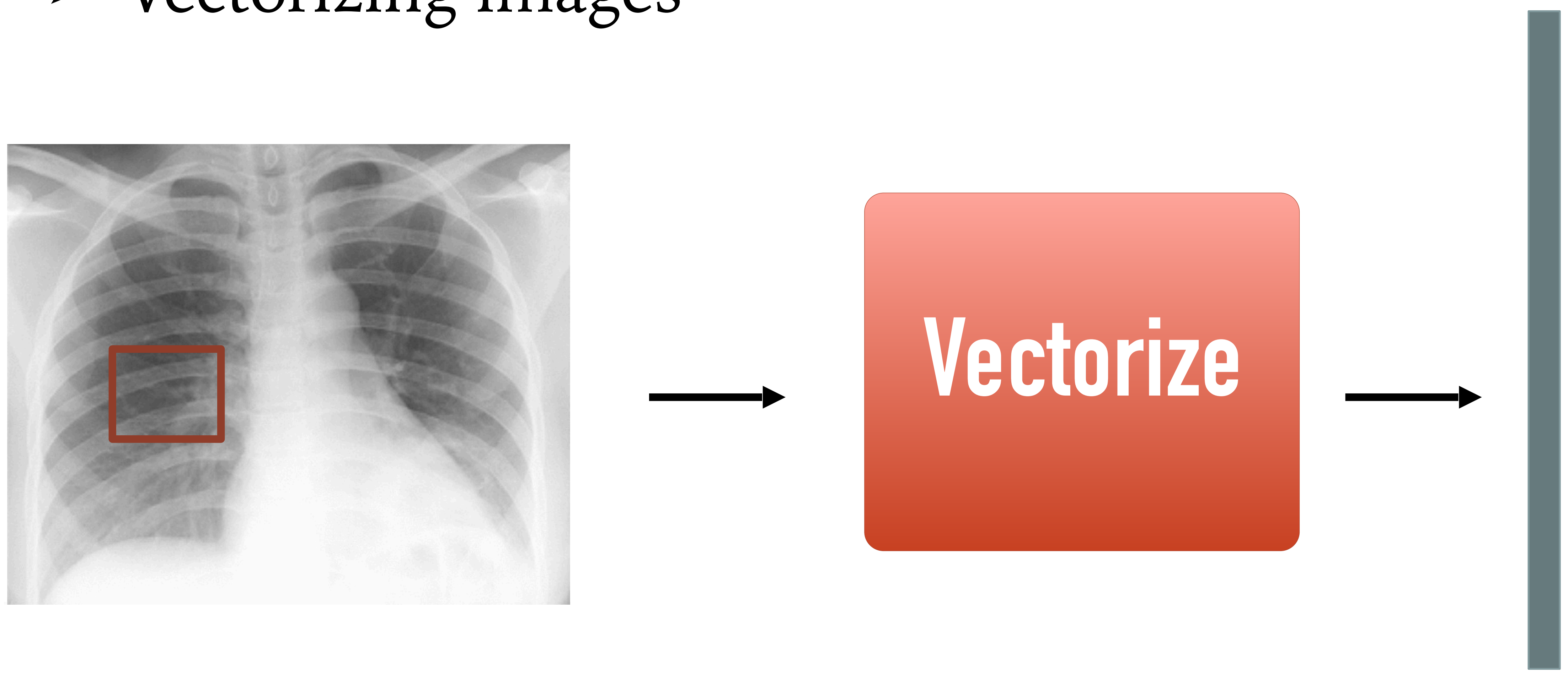
- As batch normalization is dependent on batch size, it's not effective for small batch sizes. Layer normalization is independent of the batch size, so it can be applied to batches with smaller sizes as well.

- Batch normalization requires different processing at training and inference times. As layer normalization is done along the length of input to a specific layer, the same set of operations can be used at both training and inference times.

Neural Network Architectures

WHAT MAKES DNN SUBOPTIMAL FOR IMAGES

- Vectorizing images



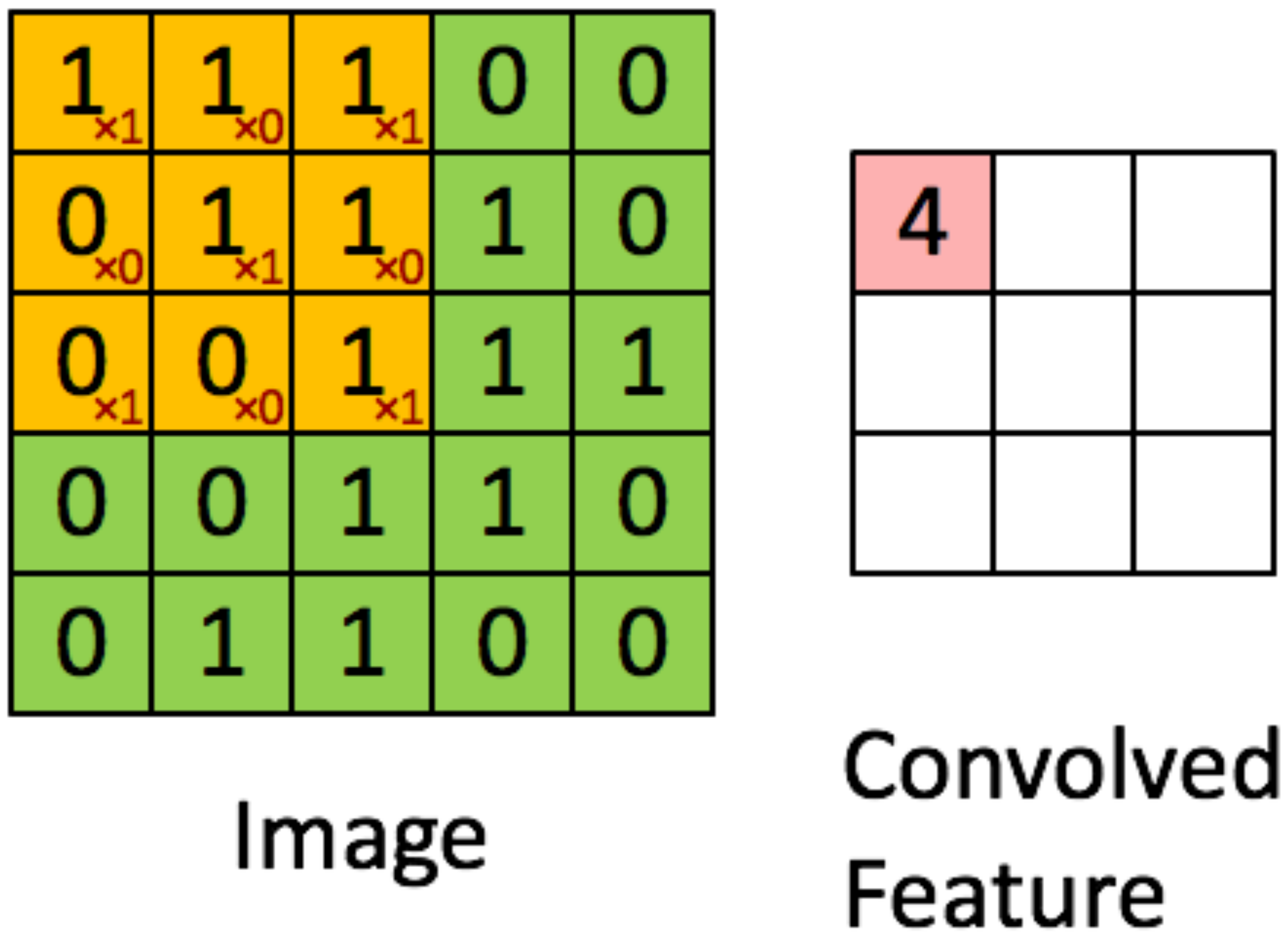
$\mathbf{x} \in \mathcal{R}^D$

- Ignores the local correlations in the pixels
  - geometric structure is not exploited in images

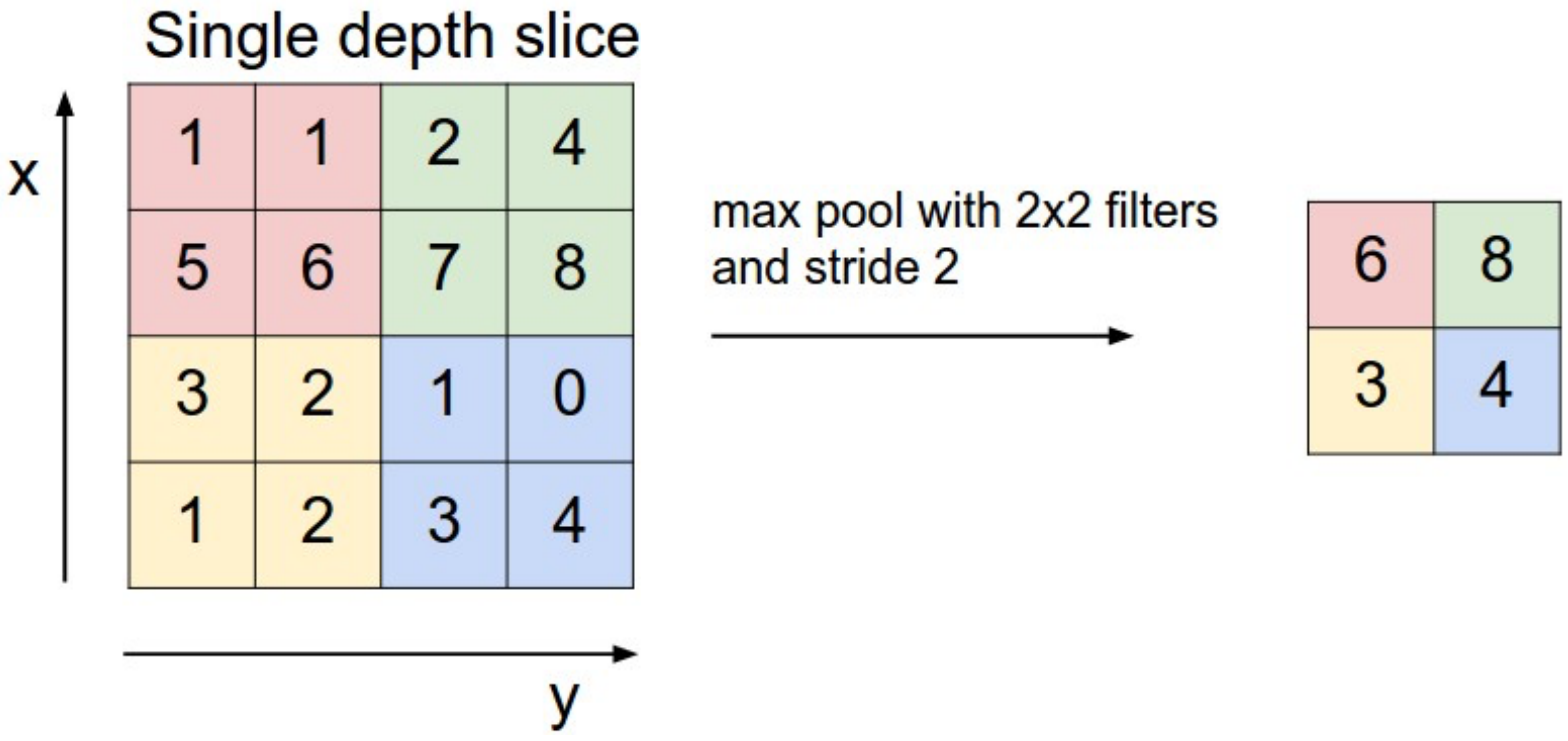
CONVOLUTIONAL NEURAL NETWORKS

➤ 2-D convolution

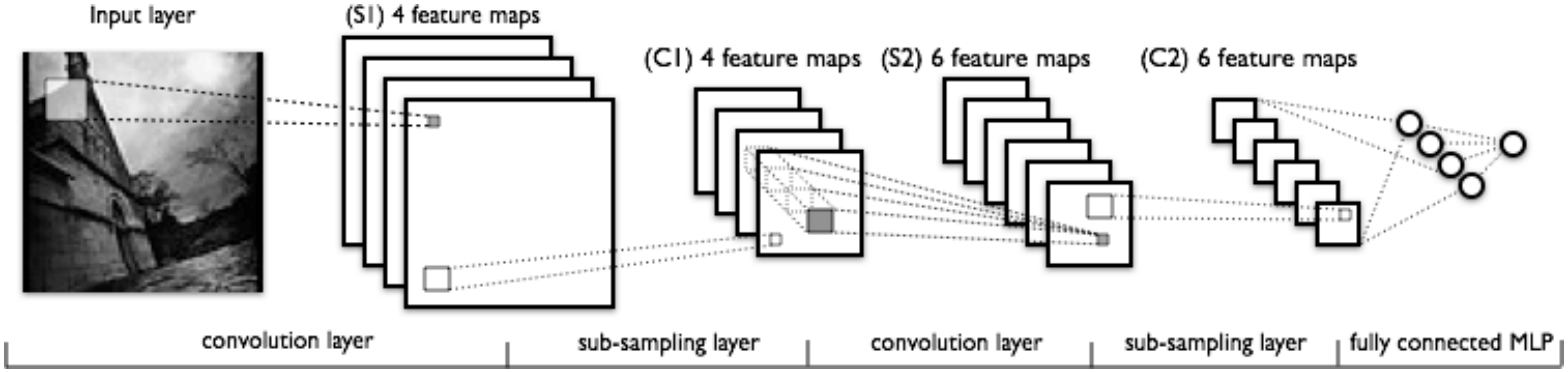
$$A^1(i,j) = \sum_{m=0}^{M} \sum_{n=0}^{N} W^1(m,n) X(i+m, j+n)$$



- Reduce the size of images after convolution using pooling
  - Keep local maximum



CONVOLUTIONAL NEURAL NETWORKS

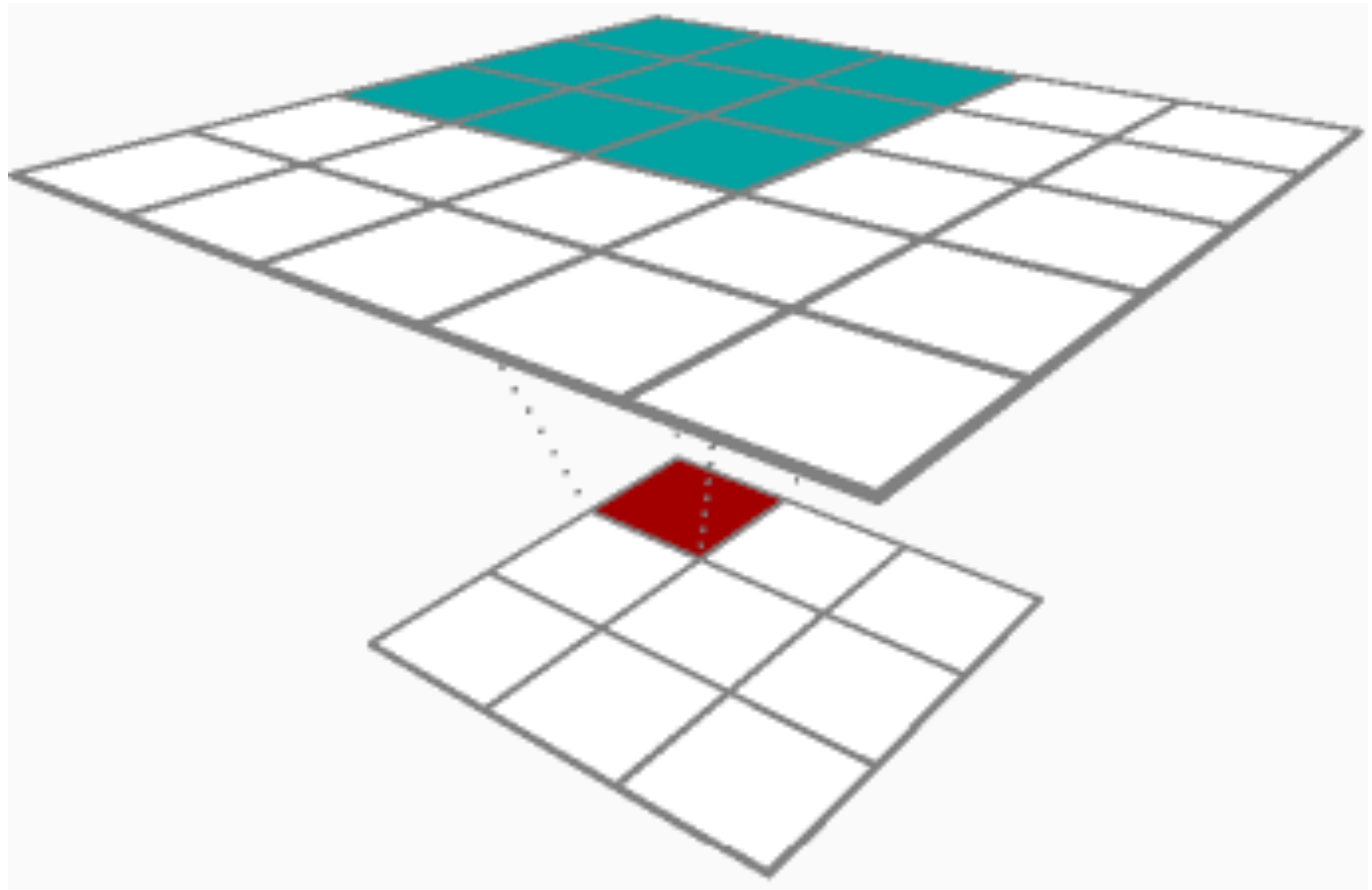


- Multiple levels of filtering and subsampling operations.
- Feature maps are generated at every layer.

BACKPROPAGATION IN CNNS

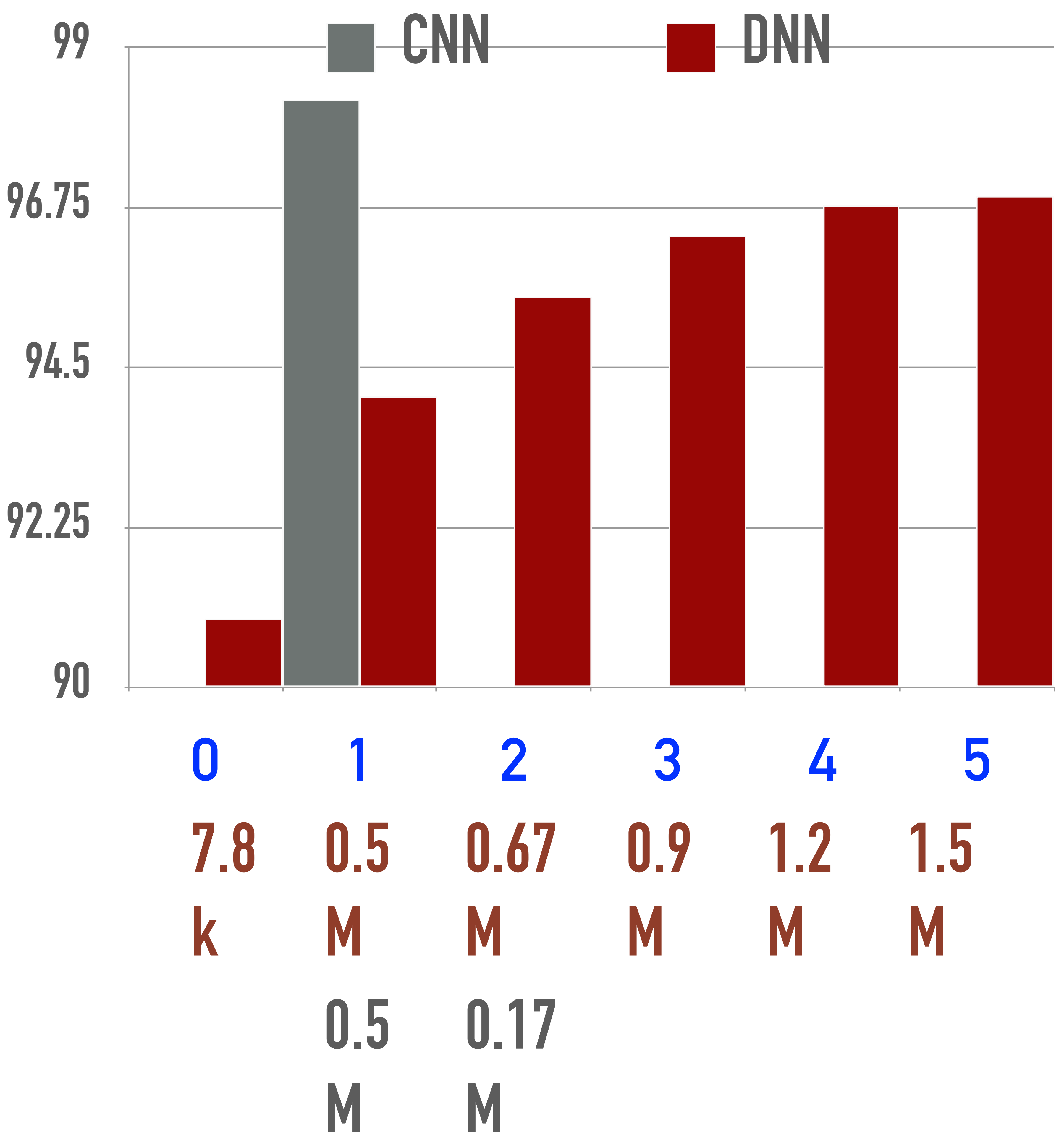
PROPERTIES OF CNN

- Reduce number of parameters
  - due to weight sharing.
  - Depth does not necessarily increase the parameter size.
- Preserving local structure
  - CNN filters operate on local weights
  - Deeper layers
    - capture wider input context.
- Training is more memory intensive
  - Accumulate gradients.



CNNS FOR MNIST

- Providing the right architecture
  - Improves the performance
  - also reduces the number of trainable parameters



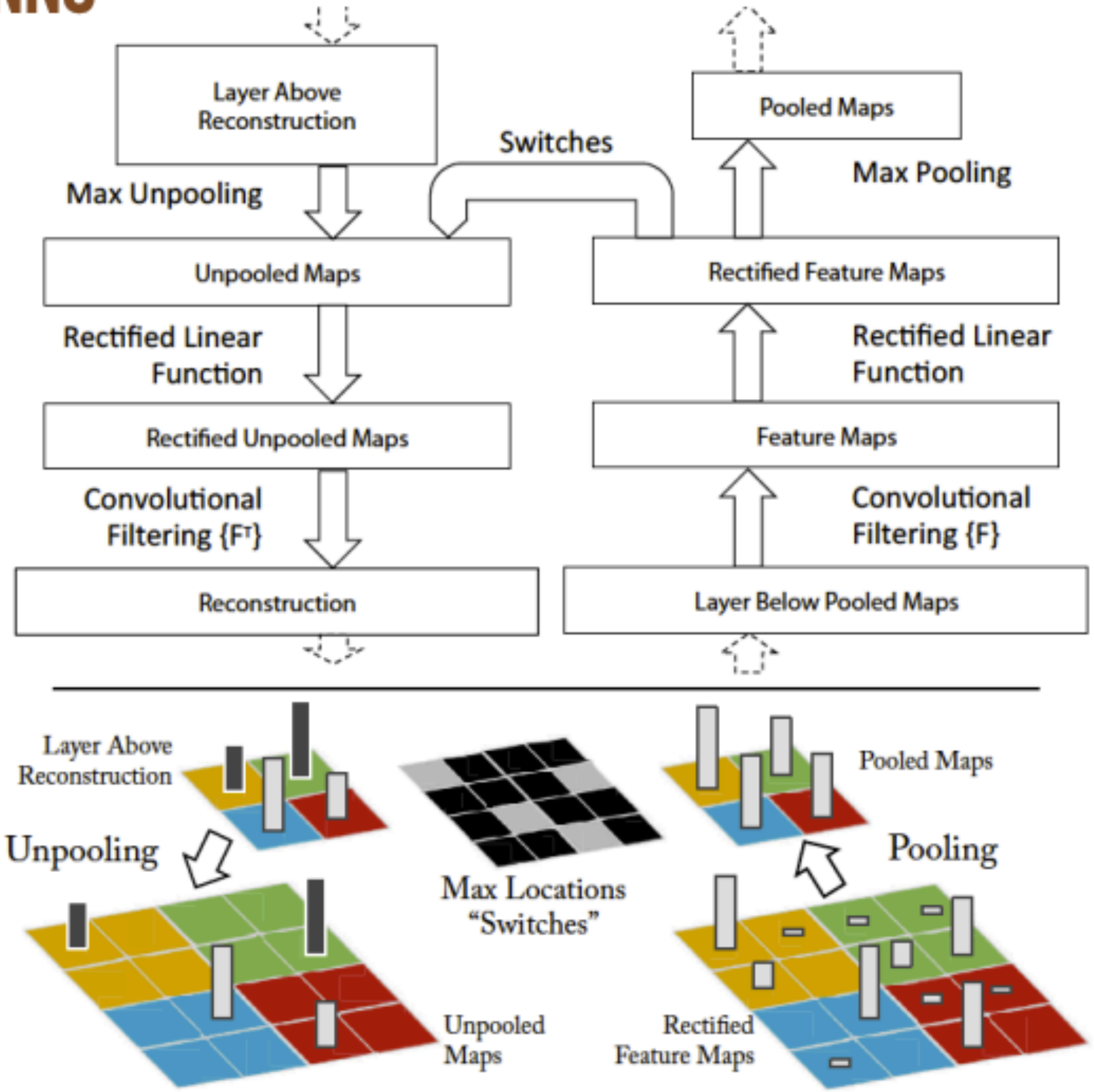
Visualizing and Understanding Convolutional Networks

Matthew D. Zeiler and Rob Fergus

Dept. of Computer Science,
New York University, USA
{zeiler,fergus}@cs.nyu.edu

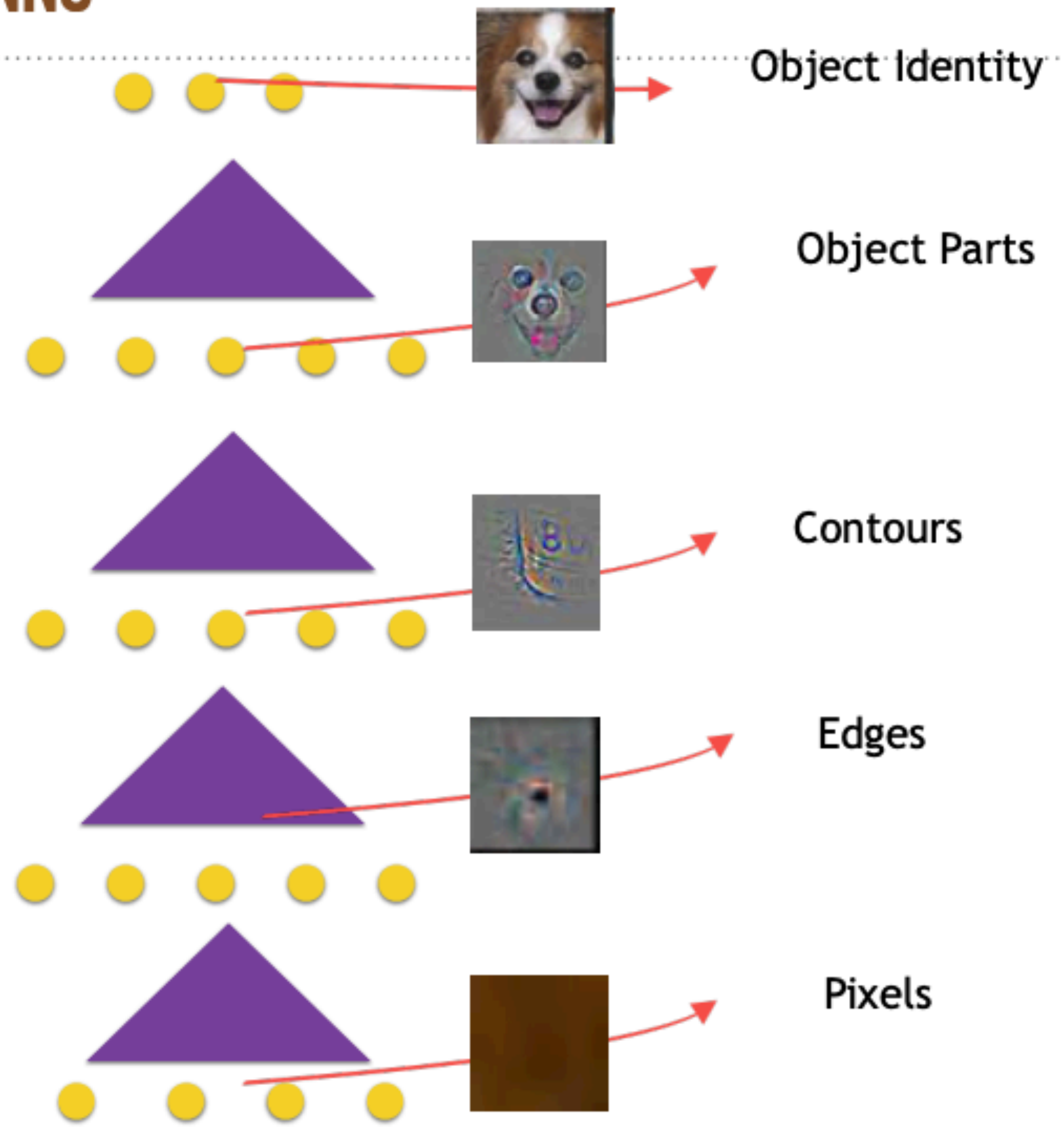
UNDERSTANDING CNNS USING USING CNNS

- Analyze a trained model
  - Take a model which is trained to perform object classification
  - Map the layer outputs back to the original space of images
  - Analyze the reconstruction outputs on a held out data



UNDERSTANDING CNNS USING USING CNNS

- Model learns a hierarchy
  - Earlier layers perform simpler tasks like edge detection.
  - Final layers perform complex tasks like object parts
- **Deep networks perform hierarchical data abstraction.**



THANK YOU

Sriram Ganapathy and TA team
LEAP lab, C328, EE, IISc
sriramg@iisc.ac.in