

# PyTorch Tutorial: Part III

Yunfei Teng

yt1208@nyu.edu

Department of Electrical and Computer Engineering  
New York University Tandon School of Engineering

March 5, 2019

**In this tutorial, some advanced models will be introduced.**

\* All the materials included in the slides are only used for the purpose of academic education.

- 1 Normalization
- 2 Autoencoder
- 3 ResNet
- 4 Skip-connection
- 5 U-Net
- 6 VisualBackProp
- 7 Generative Adversarial Nets

# Normalization

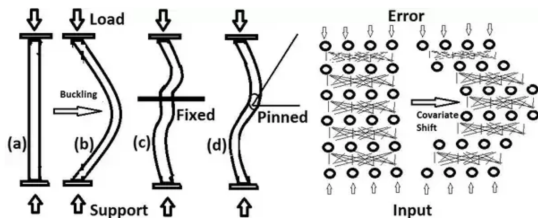
Why do we need normalization?

- Batch Norm: Reduce internal co-variate shift.
- Instance Norm: Reduce the influence of contrast.

\* Instance Norm is a special case of Batch Norm ( $batchsize = 1$ ).

**Batch Norm Paper:** <https://arxiv.org/pdf/1502.03167.pdf>

**Instance Norm Paper:** <https://arxiv.org/pdf/1607.08022.pdf>



For both, Buckling or Co-Variate Shift a small perturbation leads to a large change in the later.

Debiprasad Ghosh, PhD, Uses AI in Mechanics

Figure: Visual explanation from Quora: Co-Variate Shift

# Normalization

Ideally, all the samples are i.i.d, but it's almost impossible. That's why we need normalization.

It not only accelerates the speed of convergence, but also increases the accuracy.

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

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

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean}$$
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance}$$
$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$
$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{ scale and shift}$$

Figure: Core Algorithm of Batch Normalization

# Autoencoder

Autoencoder is useful for information compression. Training an autoencoder to minimize reconstruction error amounts to maximizing a lower bound on the **mutual information** between input and learnt representation in latent space. [Stacked Denoising Autoencoders]

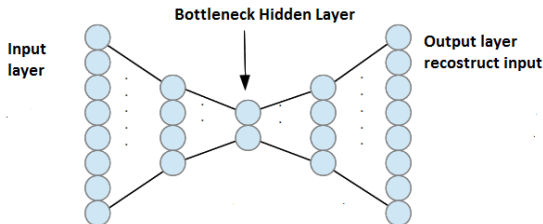


Figure: autoencoder architecture

Extended reading VAE: <https://arxiv.org/pdf/1312.6114.pdf>

**ResNet Paper:** <https://arxiv.org/pdf/1512.03385.pdf>

- Understand why the gradients are unstable. [gradient problem]
- Using residual blocks makes training arbitrarily deep neural nets become possible.

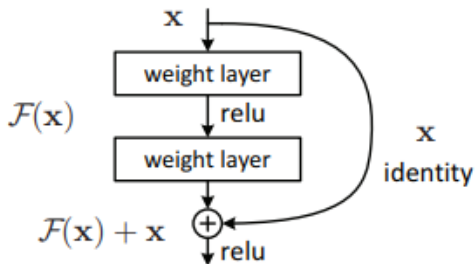
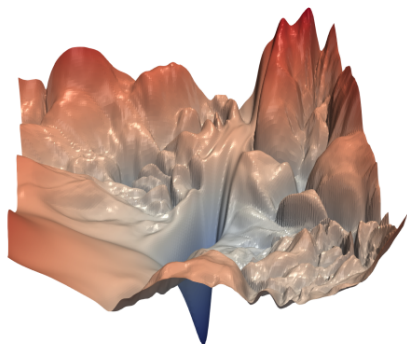


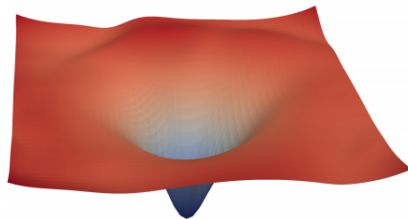
Figure: Residual block

# Skip-connection

Effect of skip-connection: <https://arxiv.org/pdf/1712.09913.pdf>



(a) without skip connections



(b) with skip connections

**Figure:** The loss surfaces of ResNet-56 with/without skip connections.

# U-Net

U-Net was first used for image segmentation but it's also a great autoencoder architecture.

**U-net Paper:** <https://arxiv.org/pdf/1505.04597.pdf>

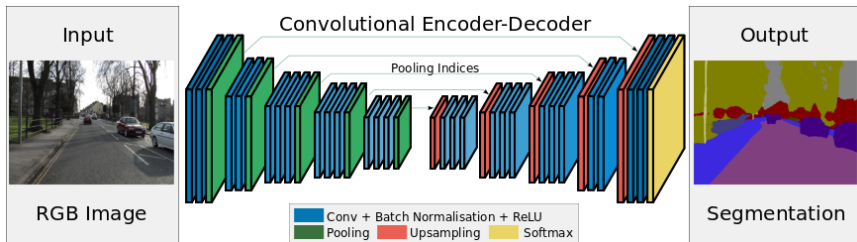


Figure: SegNet



# VisualBackProp: Efficient visualization of CNNs

**VisualBackProp Paper:** <https://arxiv.org/pdf/1611.05418.pdf>

Intuition: The feature maps contain less and less irrelevant information to the prediction decision when moving deeper into the network.

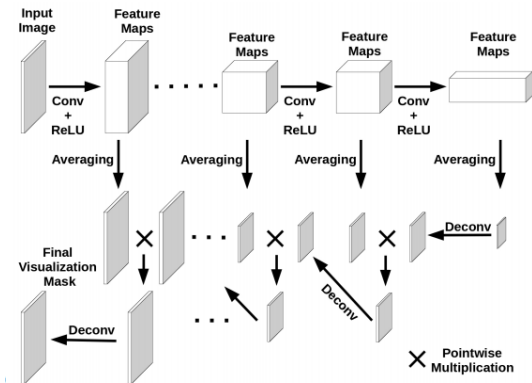


Figure: VisualBackProp

# Generative Adversarial Nets

**GAN Paper:** <https://arxiv.org/pdf/1406.2661.pdf>

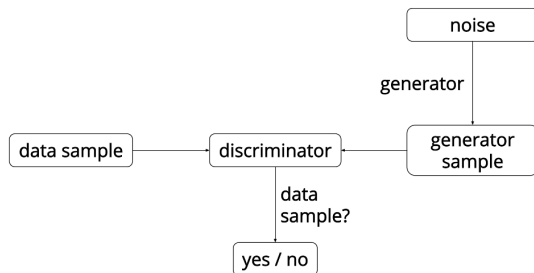


Figure: Generative adversarial nets

## Extended Reading:

- EBGAN <https://arxiv.org/pdf/1609.03126.pdf>
- Minimax: <https://en.wikipedia.org/wiki/Minimax>