Mastering MLP - Part 1

Computation Intelligence and its Application in Mechatronics

Winter 2025

Amirkabir University of Technology
(Tehran Polytechnic)

The Vanishing/Exploding Gradient Problem

- **Issue:** As gradients backpropagate, they may shrink to near-zero (vanishing) or grow too large (exploding).

- Effect:
  - Vanishing gradients slow down learning or prevent deeper layers from learning.
  - Exploding gradients cause instability and divergence.

The Vanishing/Exploding Gradient Problem

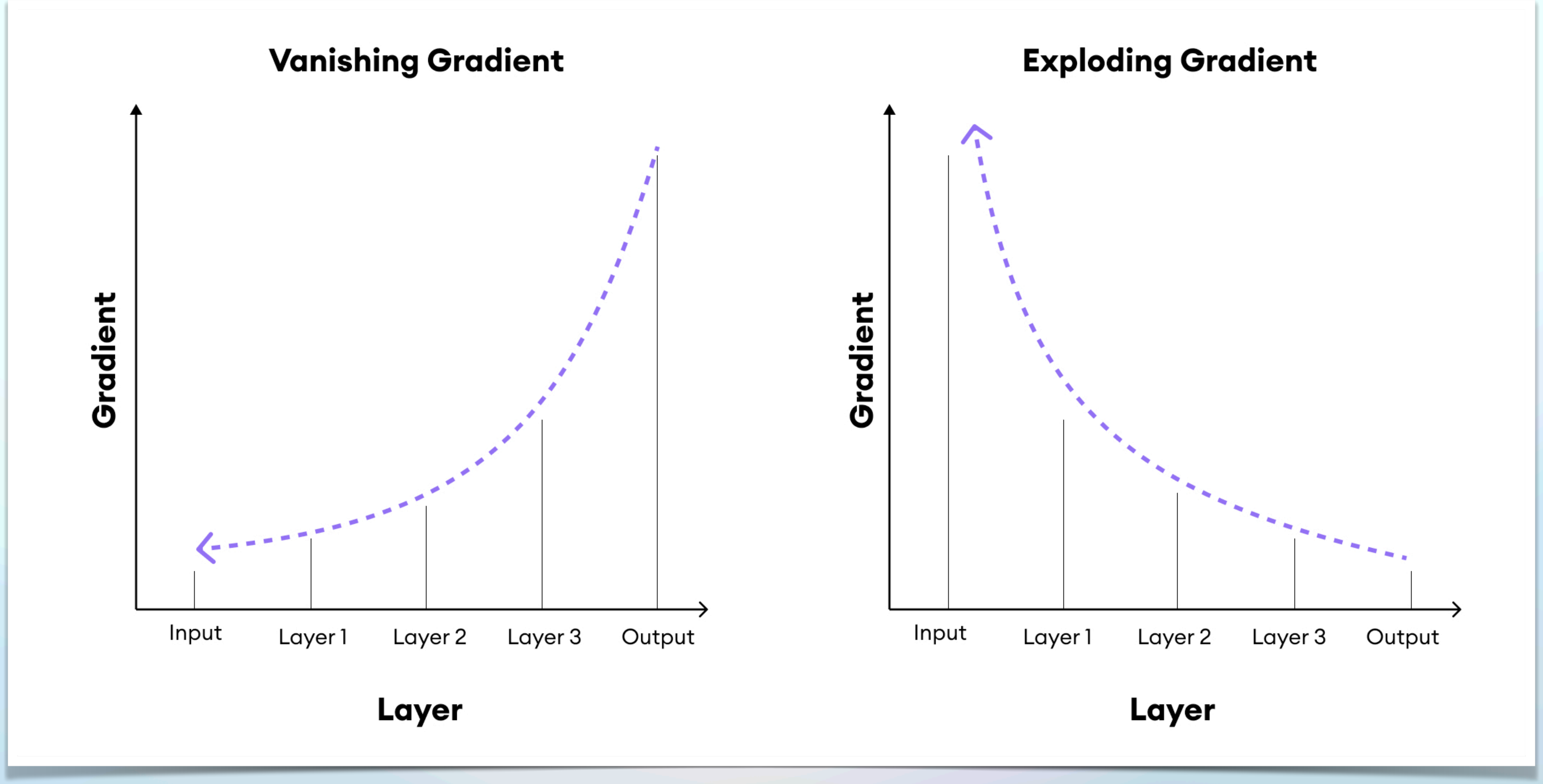


Image From: https://www.superannotate.com/blog/activation-functions-in-neural-networks

The Vanishing/Exploding Gradient Problem

Solutions to Vanishing/Exploding Gradients

1. Better Weight Initialization (Glorot, He)
2. Nonsaturating Activation Functions (Leaky ReLU, etc.)
3. Batch Normalization
4. Gradient Clipping

The Vanishing/Exploding Gradient Problem

Weight Initialization Techniques

Glorot (Xavier) Initialization

- Formula: Uniform Distribution $w \sim U\left(-\frac{\sqrt{6}}{\sqrt{n_{\text{in}} + n_{\text{out}}}}, \frac{\sqrt{6}}{\sqrt{n_{\text{in}} + n_{\text{out}}}}\right)$, Normal Distribution $w \sim \mathcal{N}\left(0, \frac{2}{n_{\text{in}} + n_{\text{out}}}\right)$
- Suitable for **sigmoid** and **tanh** activations.

He Initialization

- Formula: Uniform Distribution $w \sim U\left(-\frac{\sqrt{6}}{\sqrt{n_{\text{in}}}}, \frac{\sqrt{6}}{\sqrt{n_{\text{in}}}}\right)$, Normal Distribution $w \sim \mathcal{N}\left(0, \frac{2}{n_{\text{in}}}\right)$
- Best for **ReLU** and **Leaky ReLU** activations

The Vanishing/Exploding Gradient Problem

Nonsaturating Activation Functions

Leaky ReLU (LReLU) $= \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{if } x \leq 0 \end{cases}$

Randomized Leaky ReLU (RReLU) (randomizes α during training)

Parametric Leaky ReLU (PReLU) (learns the negative slope α during training)

Exponential Linear Unit (ELU) $= \begin{cases} x, & x > 0 \\ \alpha(\exp(x) - 1), & x \leq 0 \end{cases}$

Scaled Exponential Linear Unit (SELU)

The Vanishing/Exploding Gradient Problem

Nonsaturating Activation Functions

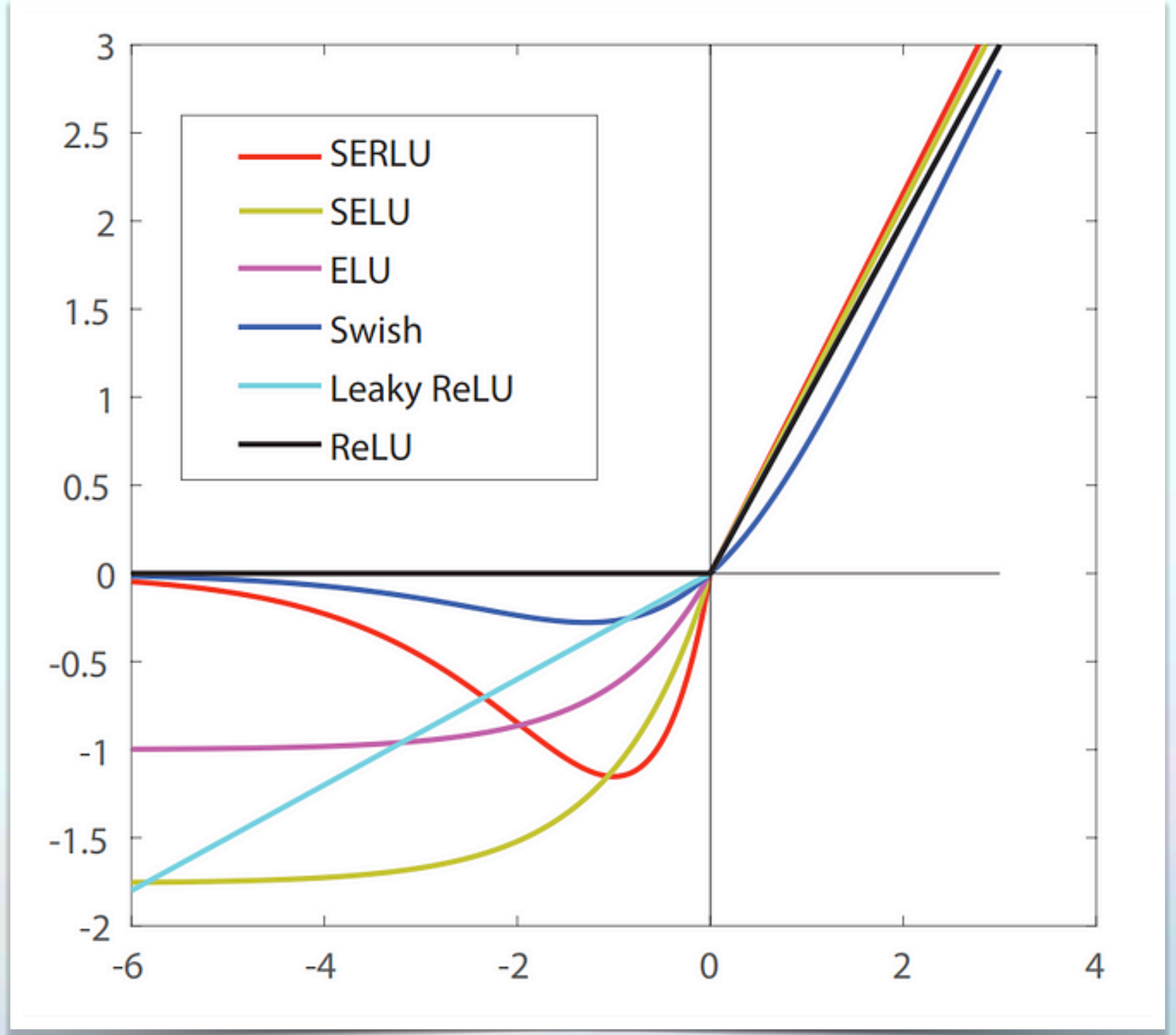


Image From: https://medium.com/towards-data-science/a-comprehensive-guide-on-activation-functions-b45ed37a4fa5

The Vanishing/Exploding Gradient Problem

Batch Normalization

Normalizes activations across the mini-batch:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \qquad z_i = \gamma \hat{x}_i + \beta$$

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i, \qquad \sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2$$

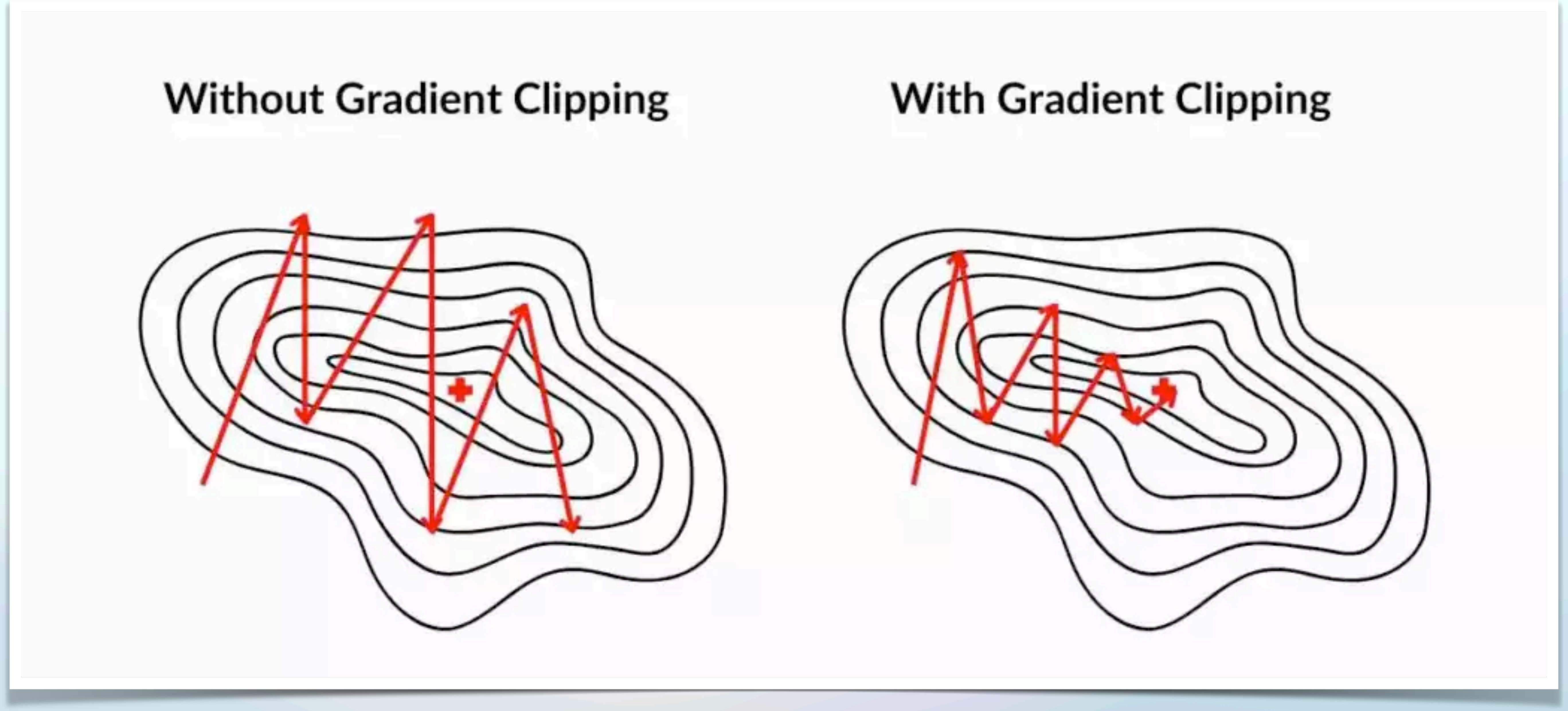
- x_i is the input activation of the neuron.
- μ_B is the mean of the mini-batch.
- σ_B^2 is the variance of the mini-batch.
- $\hat{x}_i$ is the normalized activation.
- ϵ is a small constant to prevent division by zero.
- γ (scale) and β (shift) are trainable parameters.

Image From: https://www.vecteezy.com/

The Vanishing/Exploding Gradient Problem

Gradient Clipping

Prevents exploding gradients by limiting their magnitude.



mage From: https://spotintelligence.com/2023/12/01/gradient-clipping-explained-practical-how-to-guide-in-python/

Avoiding Overfitting – Regularization Techniques

L1 and L2 regularization

L1 regularization: lasso regression

$$\text{New Cost} = \frac{1}{n}\sum_{n}^{i} y_i - \hat{y}_i + \lambda \sum_{i} |w_i|$$

L2 regularization: ridge regression

$$\text{New Cost} = \frac{1}{n}\sum_{n}^{i} y_i - \hat{y}_i + \lambda \sum_{i} w_i^2$$

Avoiding Overfitting – Regularization Techniques

Dropout

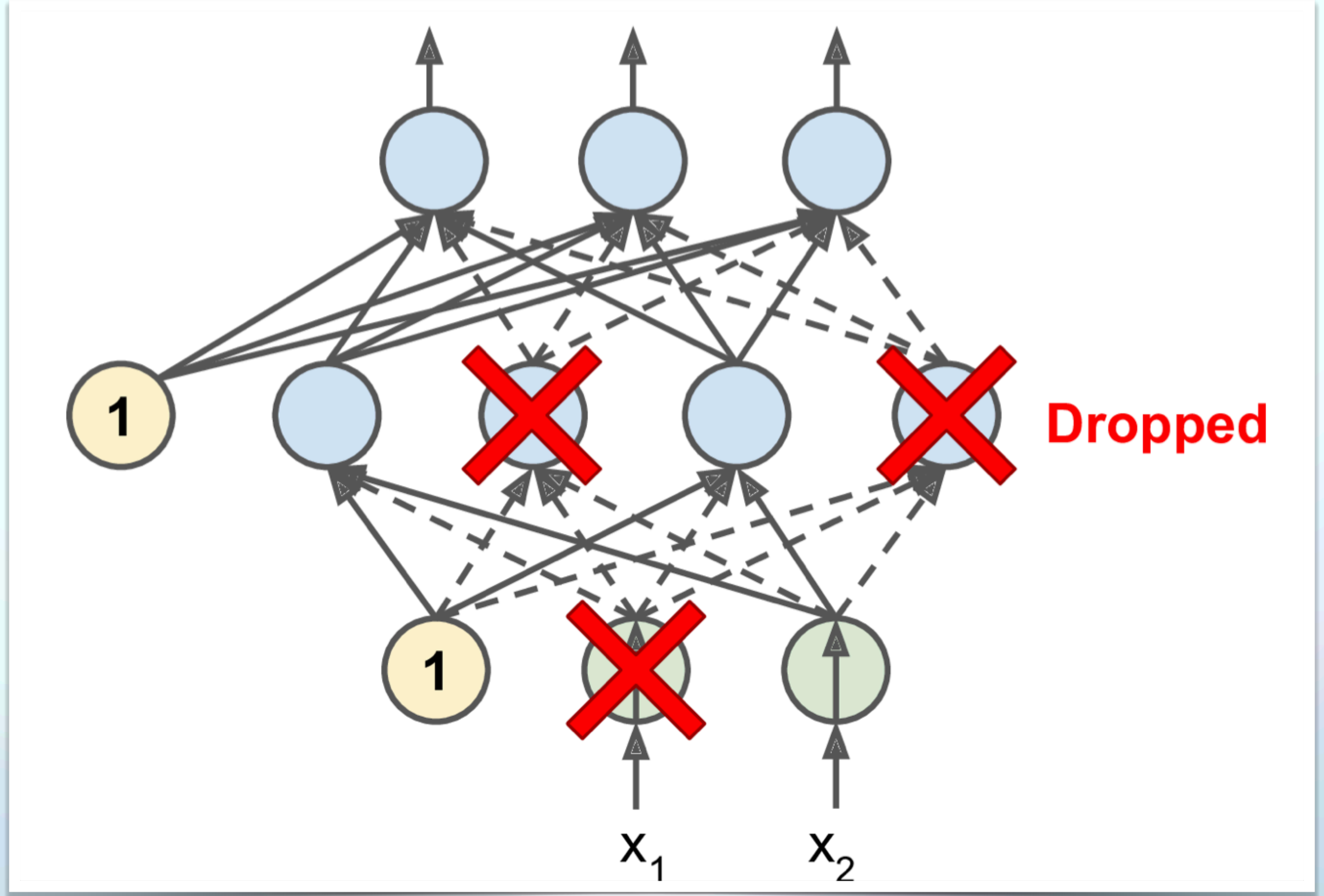


Image from: Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow

What's Coming Next

- Training your model with faster optimizers