(Artificial) Neural Networks: Advanced

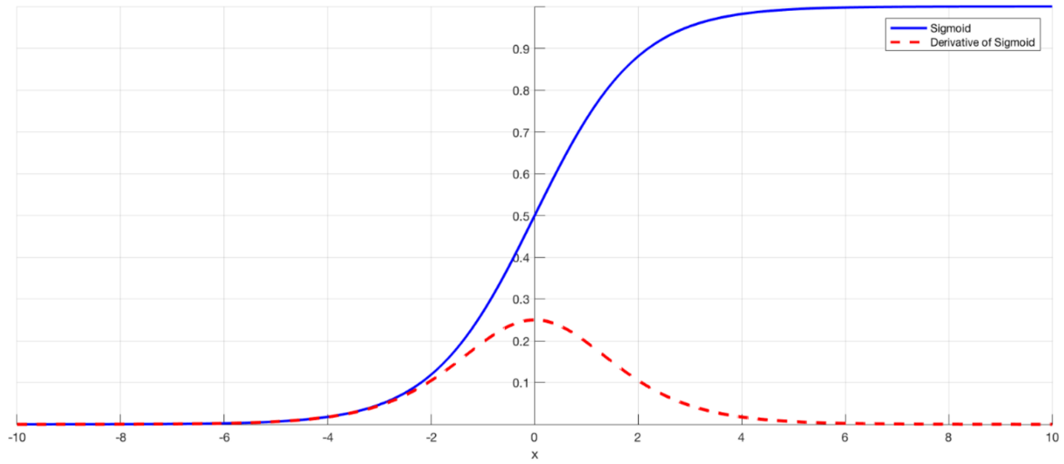
Industrial AI Lab.

Prof. Seungchul Lee

Nonlinear Activation Function

The Vanishing Gradient Problem

- As more layers using certain activation functions are added to neural networks, the gradients of the loss function approaches zero, making the network hard to train.
- For example,
  - $\frac{dz}{du} = \frac{dz}{dy} \cdot \frac{dy}{dx} \cdot \frac{dx}{dw} \cdot \frac{dw}{du}$



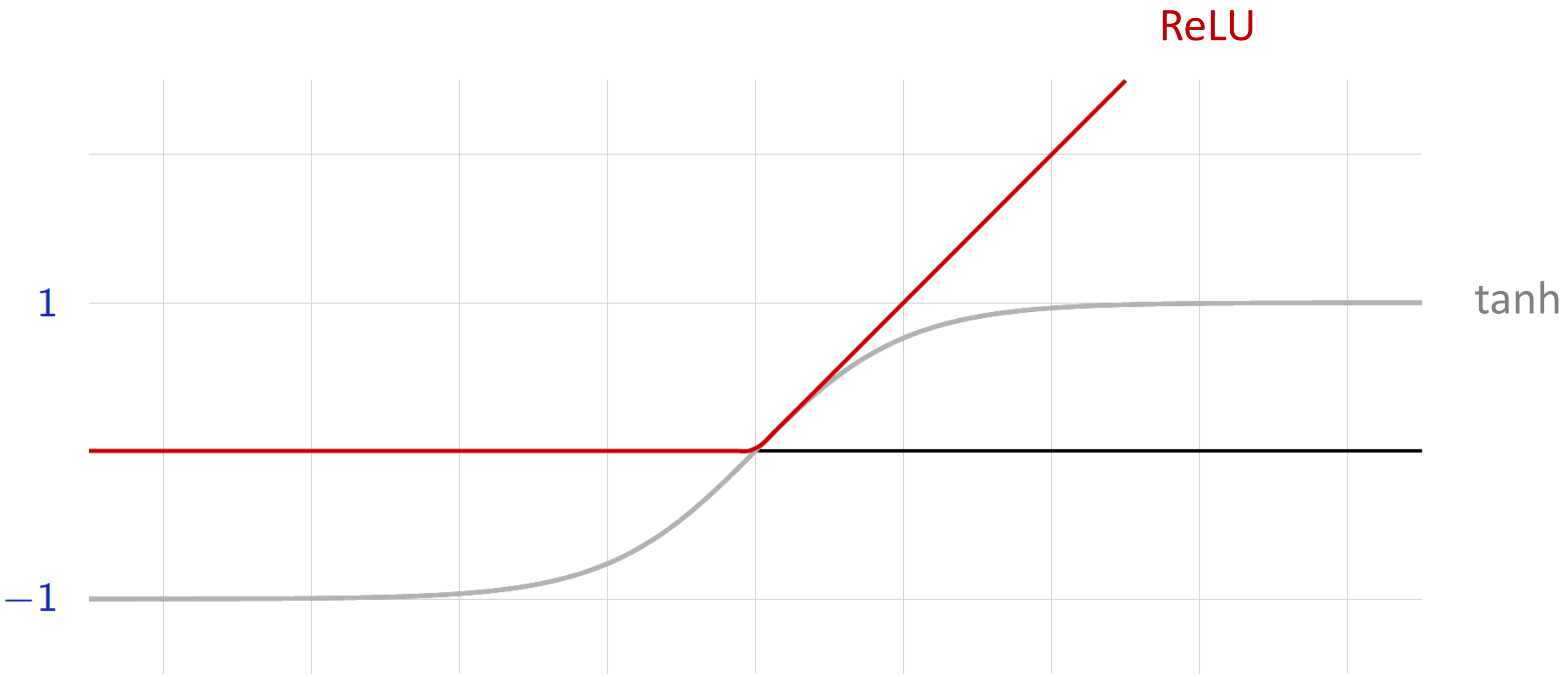
sigmoid

Derivative of sigmoid

POSTECH

Rectifiers

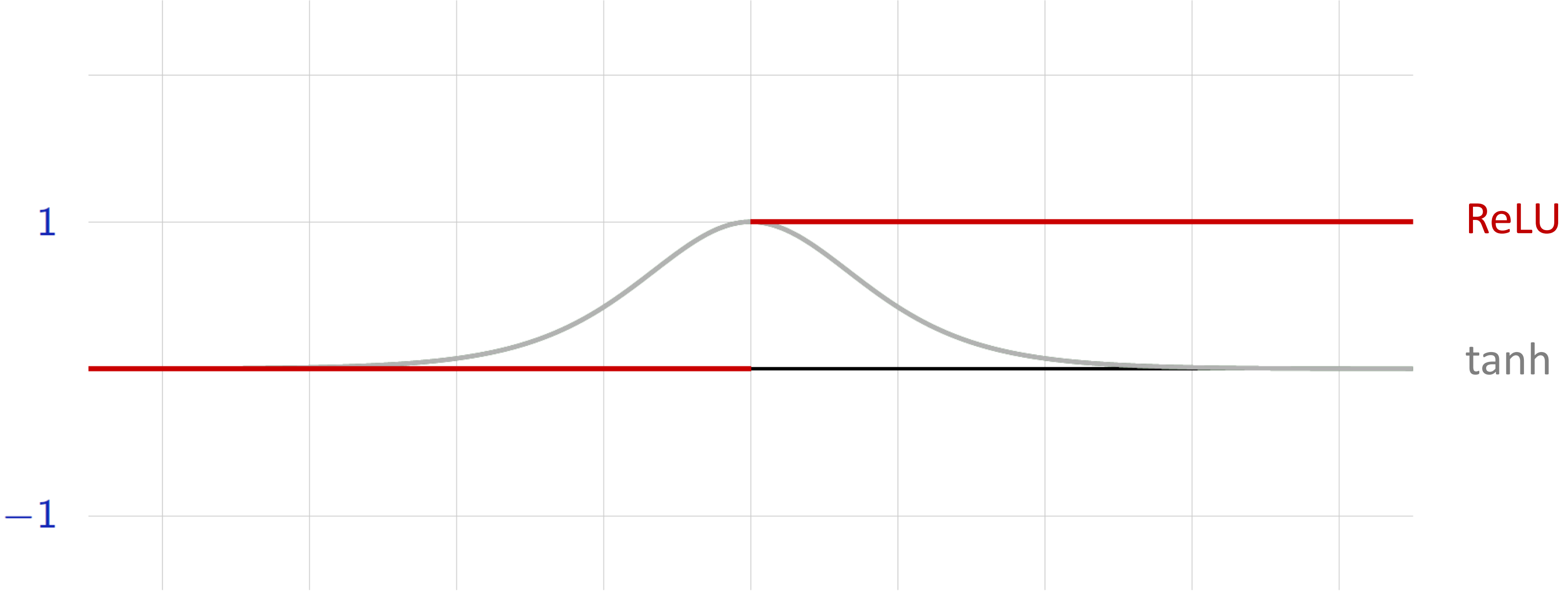
- The use of the ReLU activation function was a great improvement compared to the historical tanh.



Source: Dr. Francois Fleuret at EPFL

Rectifiers

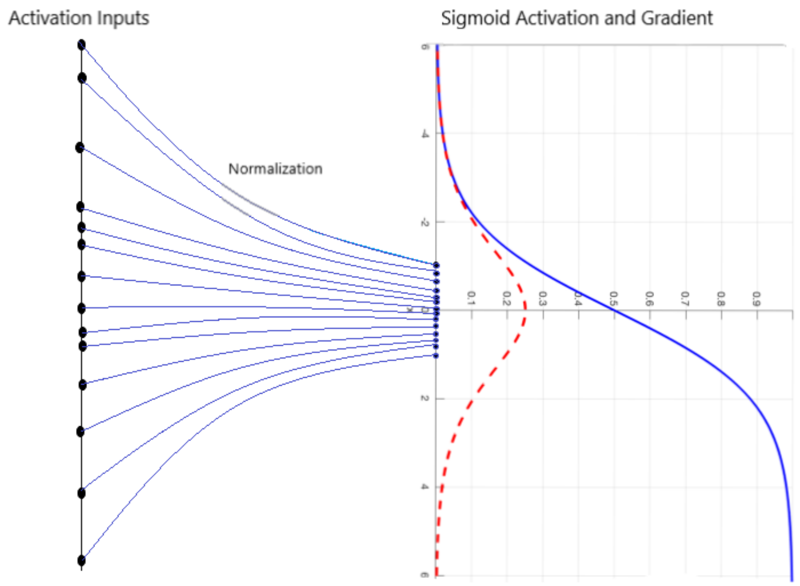
- This can be explained by the derivative of ReLU itself not vanishing, and by the resulting coding being sparse (Glorot et al., 2011).

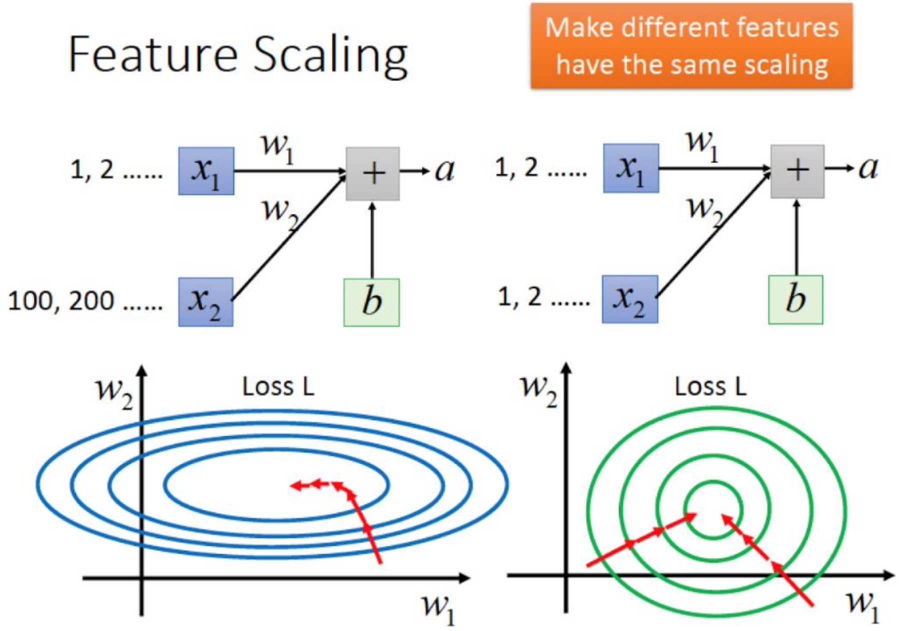


Batch Normalization

Batch Normalization

- Batch normalization is a technique for improving the performance and stability of artificial neural networks.
- It is used to normalize the input layer by adjusting and scaling the activations.





S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning (ICML), 2015.

Batch Normalization

- During training batch normalization shifts and rescales according to the mean and variance estimated on the batch.

- During test, it simply shifts and rescales according to the empirical moments estimated during training.

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.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning (ICML), 2015.

Dropout as Regularization

Regularization (Shrinkage Methods)

- Often, overfitting associated with very large estimated parameters
- We want to balance
  - how well function fits data
  - magnitude of coefficients

$$\text{Total cost} = \underbrace{\text{measure of fit}}_{RSS(\theta)} + \; \lambda \cdot \underbrace{\text{measure of magnitude of coefficients}}_{\lambda \cdot \|\theta\|_2^2}$$

$$\implies \min \; \|\Phi\theta - y\|_2^2 + \lambda\|\theta\|_2^2$$

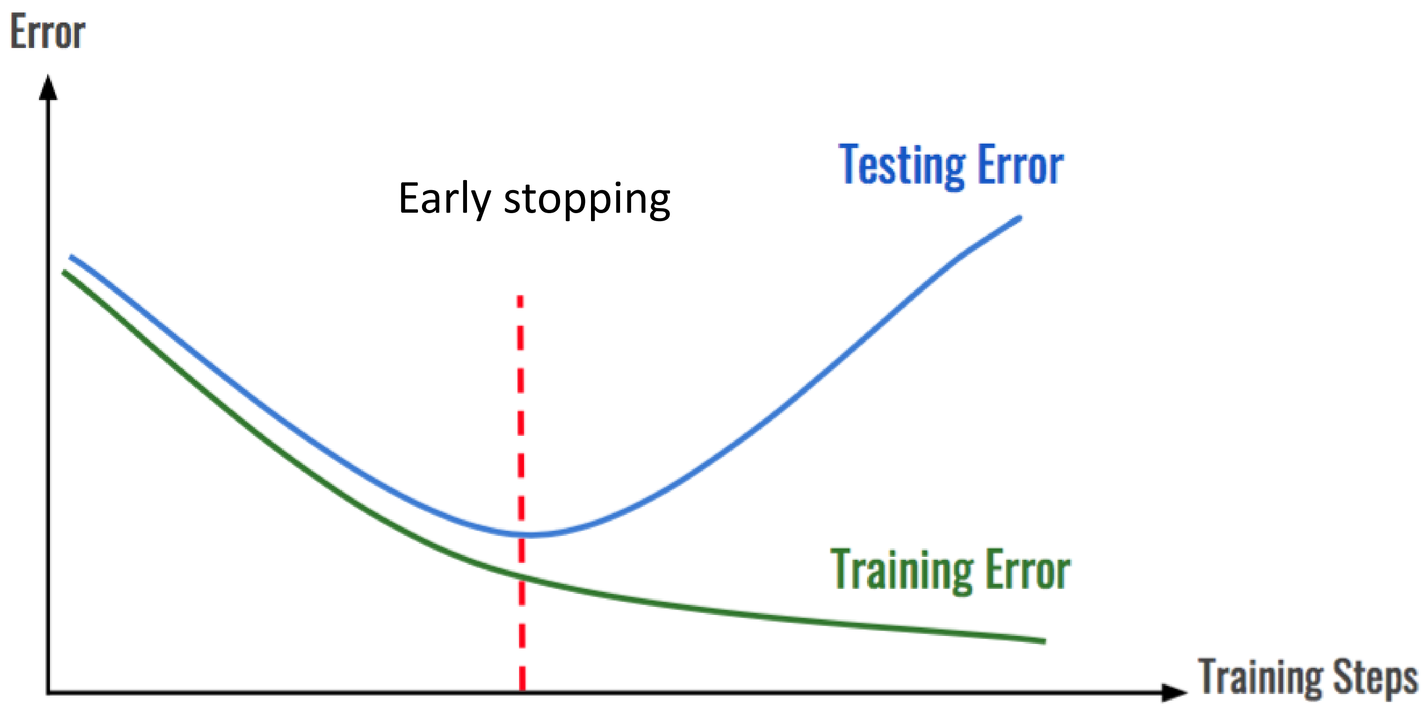
  - multi-objective optimization
  - λ is a tuning parameter

Different Regularization Techniques

- Big Data
- Data augmentation
  - The simplest way to reduce overfitting is to increase the size of the training data.

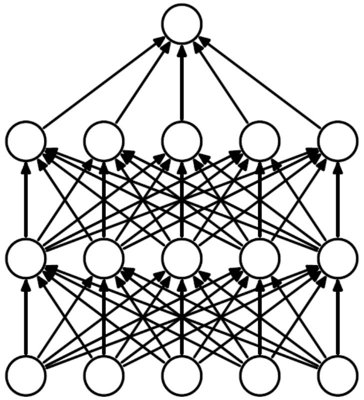
Different Regularization Techniques

- Early stopping
  - When we see that the performance on the validation set is getting worse, we immediately stop the training on the model.

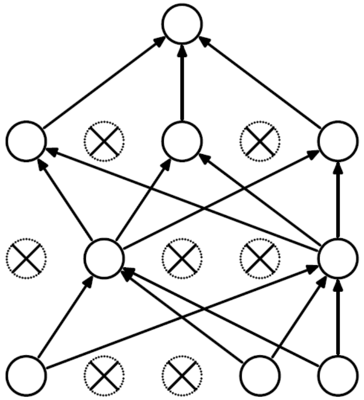


Different Regularization Techniques in Deep Learning

- Dropout
  - This is the one of the most interesting types of regularization techniques.
  - It also produces very good results and is consequently the most frequently used regularization technique in the field of deep learning.
  - At every iteration, it randomly selects some nodes and removes them.
  - It can also be thought of as an ensemble technique in machine learning.



(a) Standard Neural Net

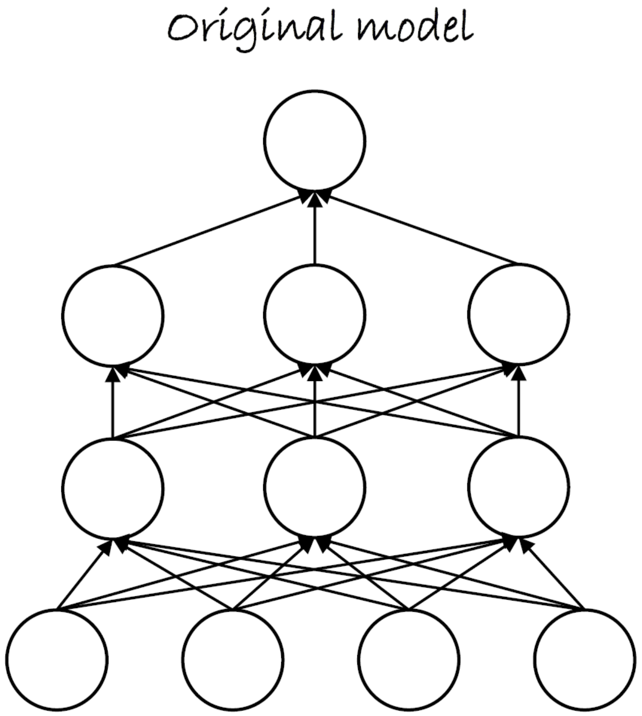


(b) After applying dropout.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research (JMLR), 15:1929-1958, 2014.

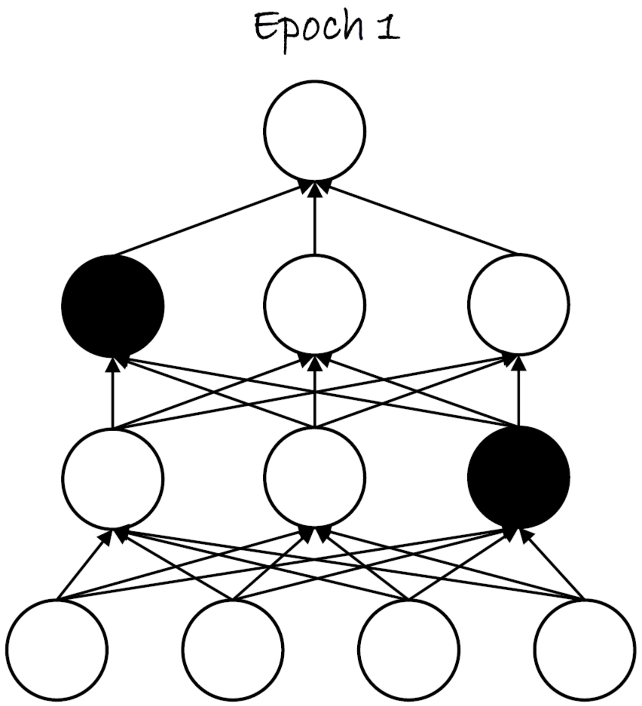
Dropout Illustration

- Effectively, a different architecture at every training epoch
- It can also be thought of as an ensemble technique in machine learning.



Dropout Illustration

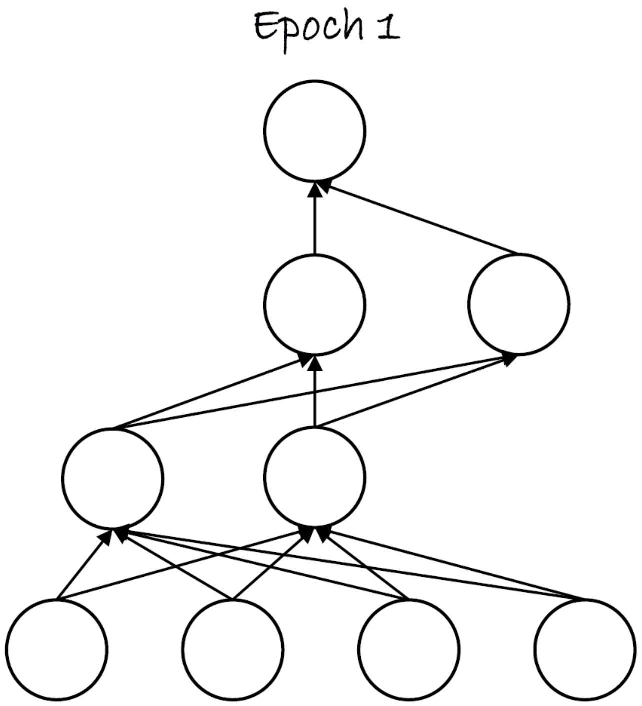
- Effectively, a different architecture at every training epoch
- It can also be thought of as an ensemble technique in machine learning.



tf.nn.dropout(layer, keep_prob = p)

Dropout Illustration

- Effectively, a different architecture at every training epoch
- It can also be thought of as an ensemble technique in machine learning.



tf.nn.dropout(layer, keep_prob = p)