Idea Factory Intensive Program #2

딥러닝 홀로서기

#13

이론강의/PyTorch실습/코드리뷰

딥러닝(Deep Learning)에 관심이 있는 학생 발굴을 통한
딥러닝의 이론적 배경 강의 및 오픈소스 딥러닝 라이브러리 PyTorch를 활용한 실습

Topics to learn today

1. Review from last lecture

Assignment: MNIST classification with Pytorch

Lecture: Classification and MLP

2. Why MLP does not work?

Overfitting: L2 norm and Dropout

Gradient Vanishing: Activation functions

3. Other techniques

Batch Normalization

Xavier Initialization

Review from Last Lecture

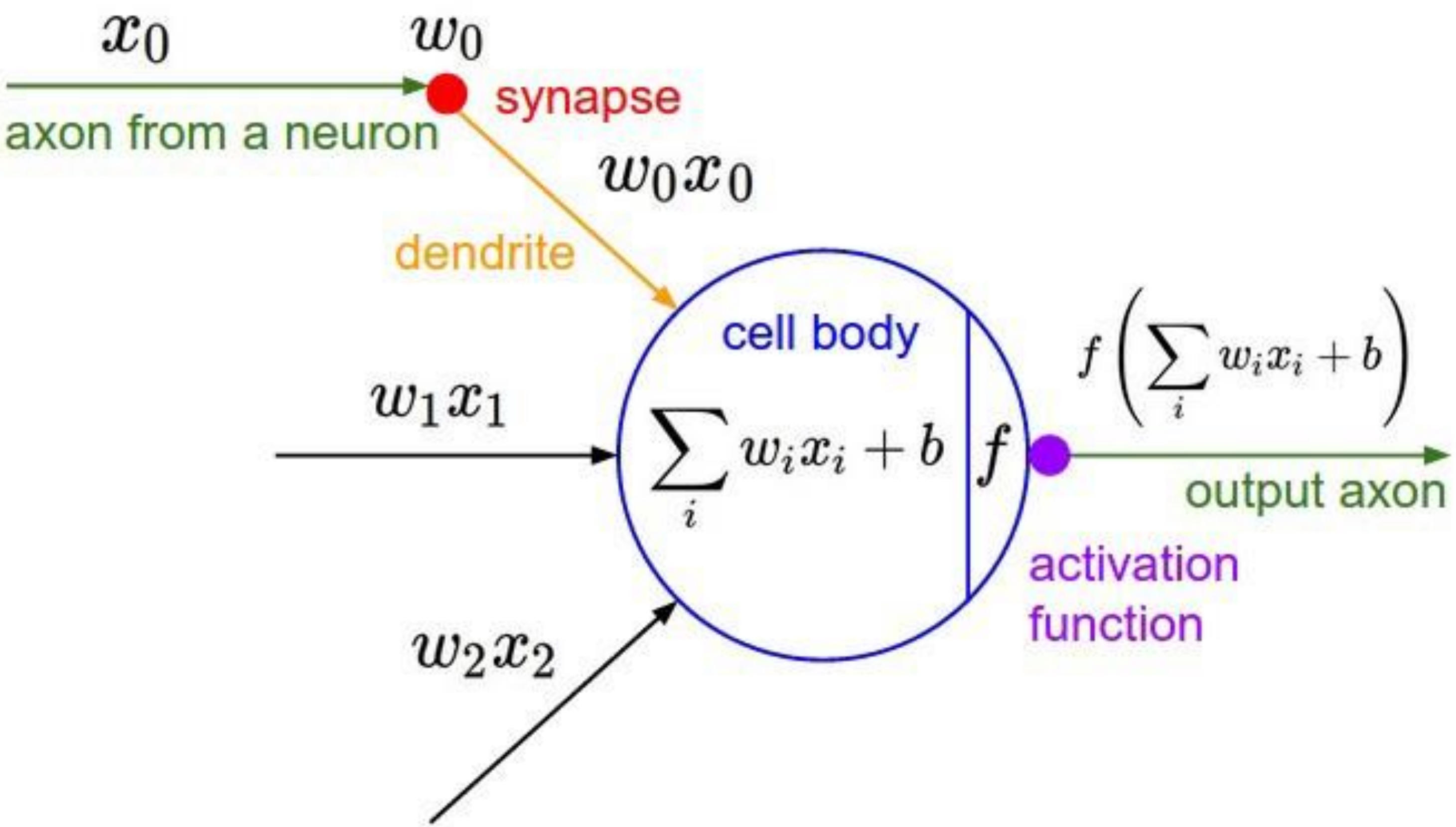
	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Discrete	Classification	Clustering	Discrete Action Space Agent
Continuous	Regression	Dimensionality Reduction	Continuous Action Space Agent

Semi-Supervised Learning

Binary and Multinomial Classification

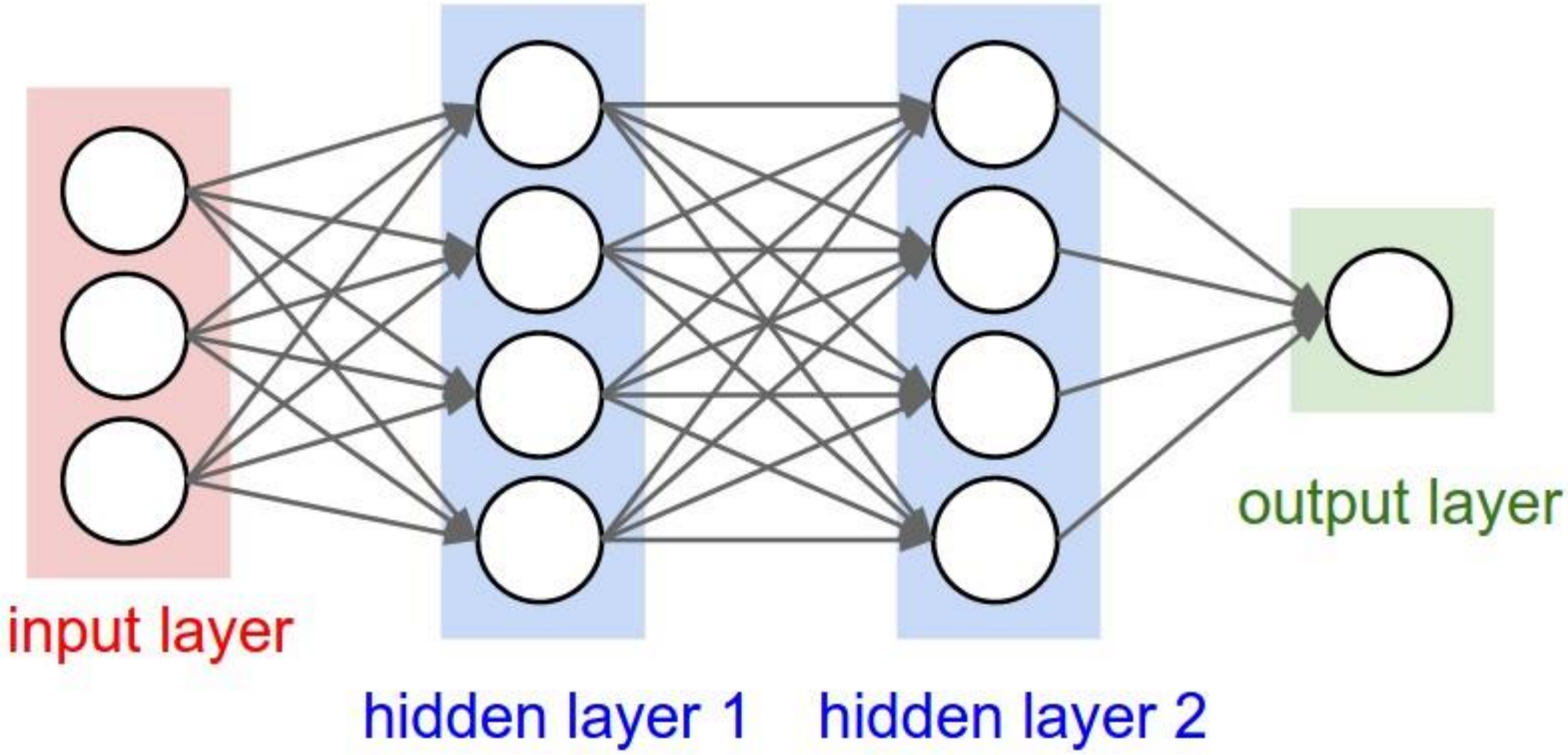
	Binary	Multinomial
Hypothesis	$Sigmoid(WX)$	$Softmax(WX)$
Cost	$-ylog(H(x)) - (1-y)log(1-H(x))$	$\sum -ylog(H(X))$

Modeling Neuron (1957)



Stanford, cs231n 2017

MLP Structure



https://beamandrew.github.io/deeplearning/2017/02/23/deep_learning_101_part2.html

Problems of MLP
Overfitting

Model Capacity

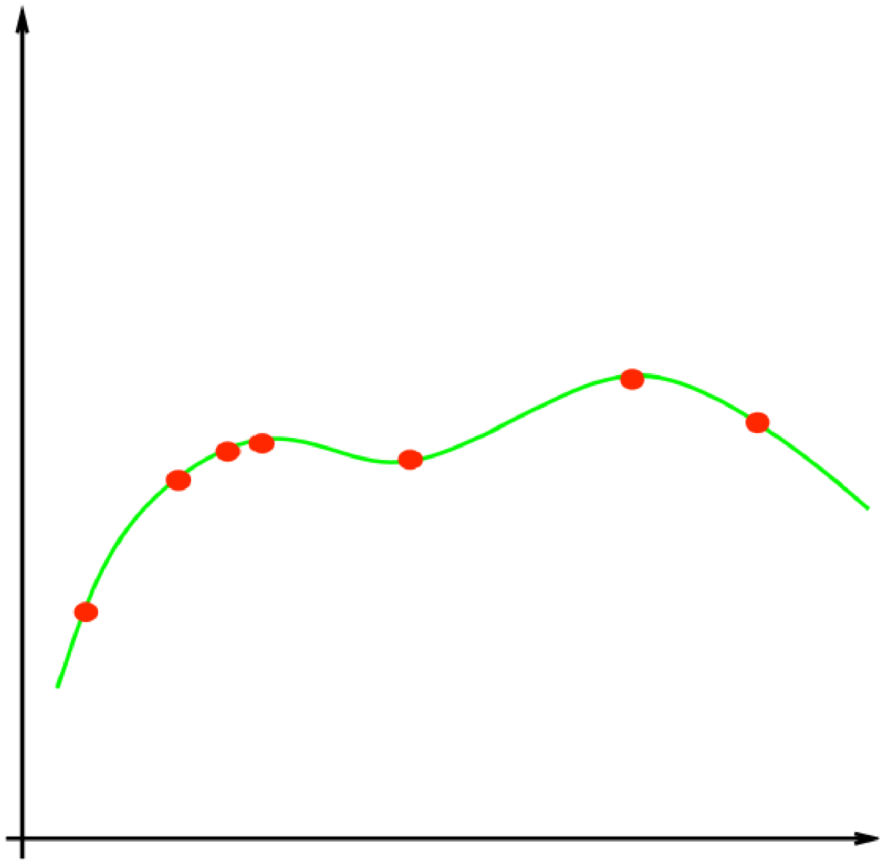
Overfitting

Overfitting

True Risk

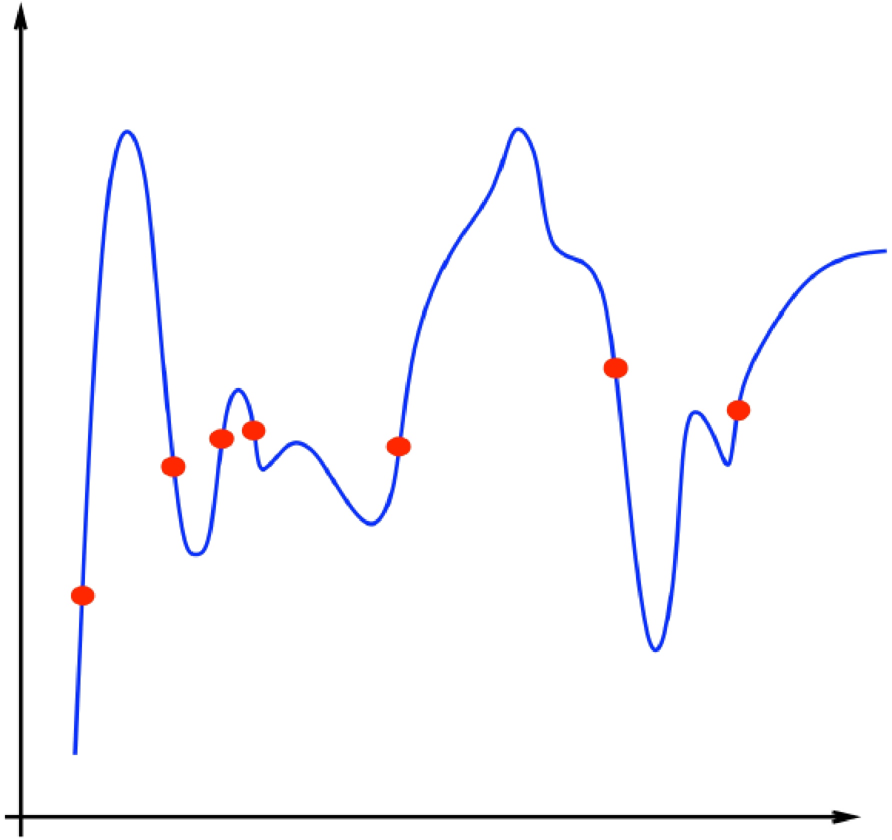
Empirical Risk

Overfitting



[T. Poggio]

Overfitting



[T. Poggio]

Overfitting

Summary

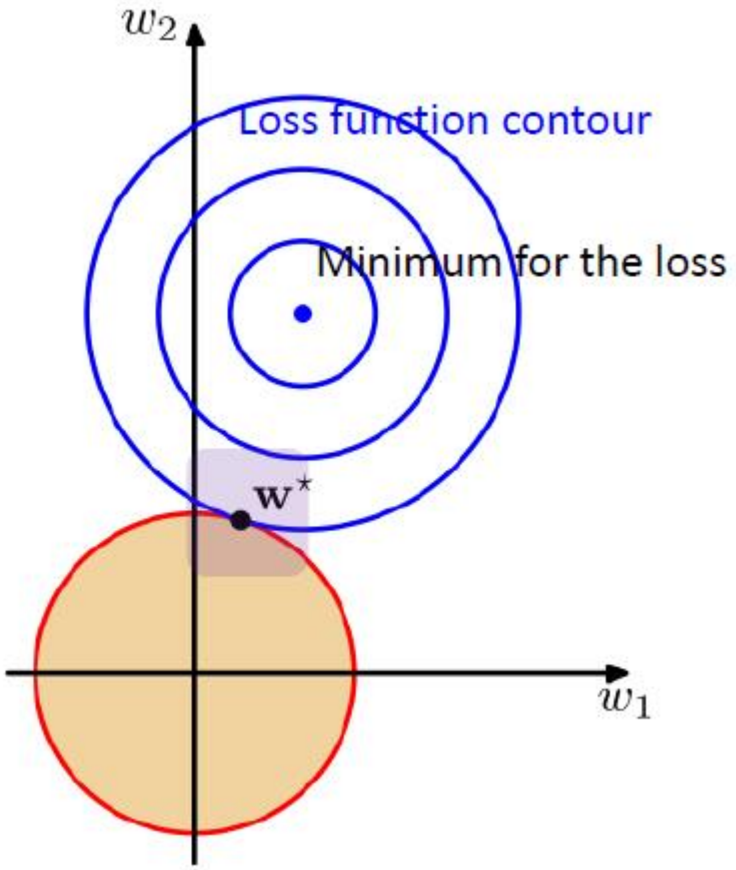
Overfitting occurs when model memorizes the train dataset thus cannot be applied to general dataset such as test set.

As model capacity increases, model become available to represent complicated systems but also be able to memorize the specific dataset.

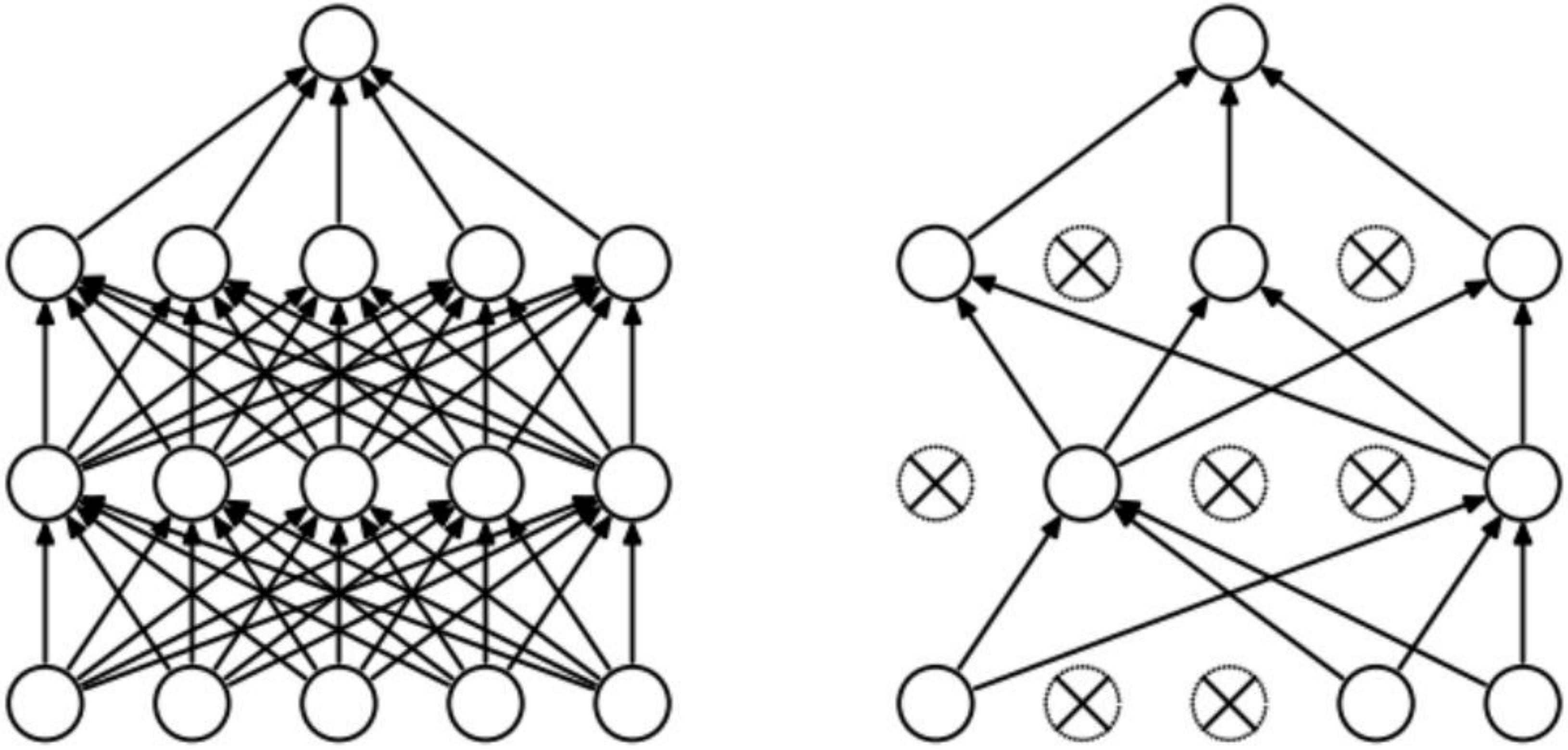
We can know whether overfitting occurs or not by comparing training loss (empirical risk) and validation loss (approximated true risk)

Regularizations
L2 Regularization and Dropout

L2 Regularization



Dropout



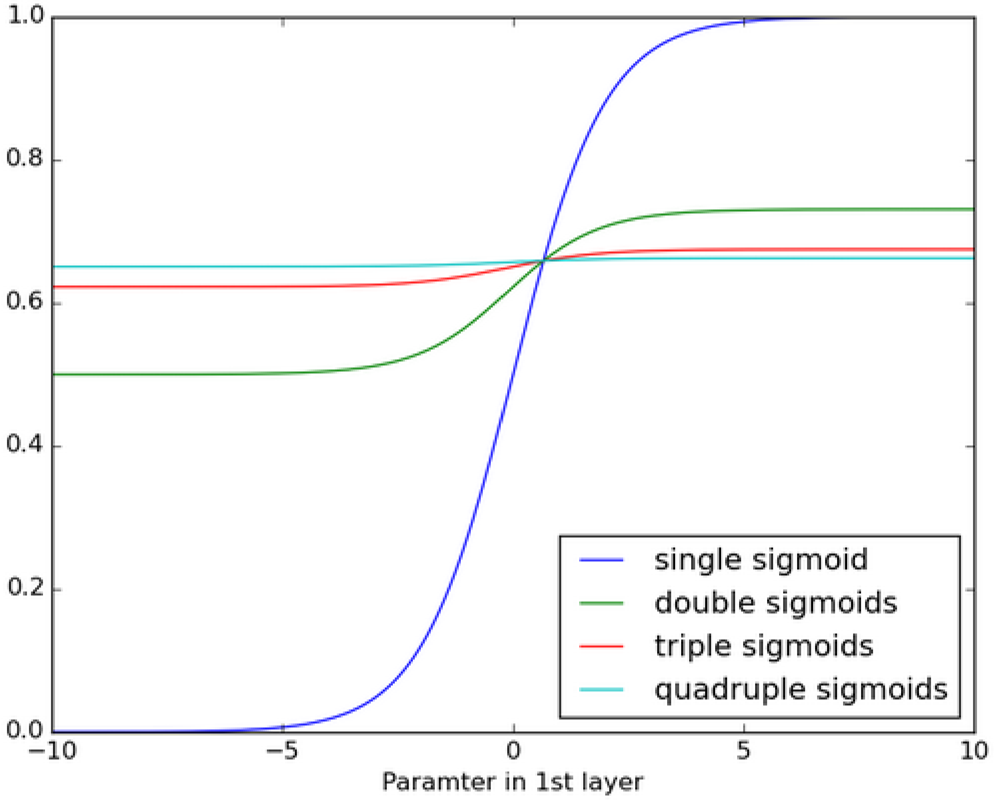
(a) Standard Neural Net　　(b) After applying dropout.

Srivastava, Nitish, et al. " Dropout: a simple way to prevent neural networks from overfitting" , JMLR 2014

Problems of MLP
Gradient Vanishing

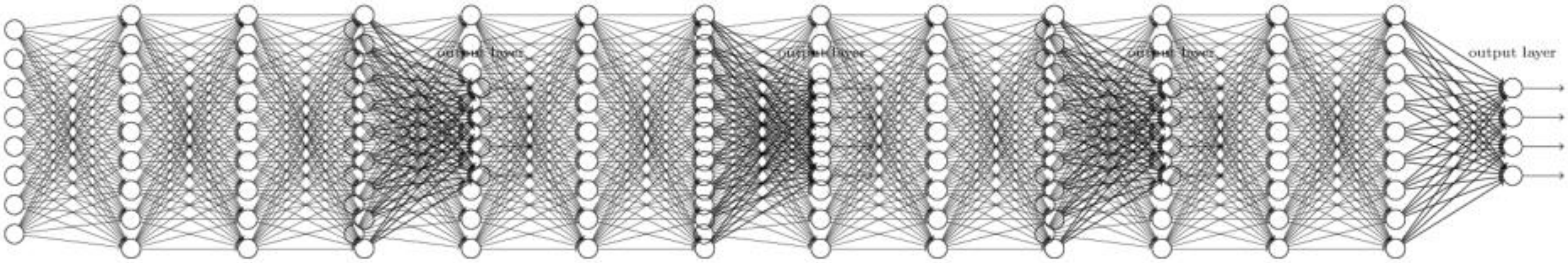
Gradient Vanishing

Gradient Vanishing

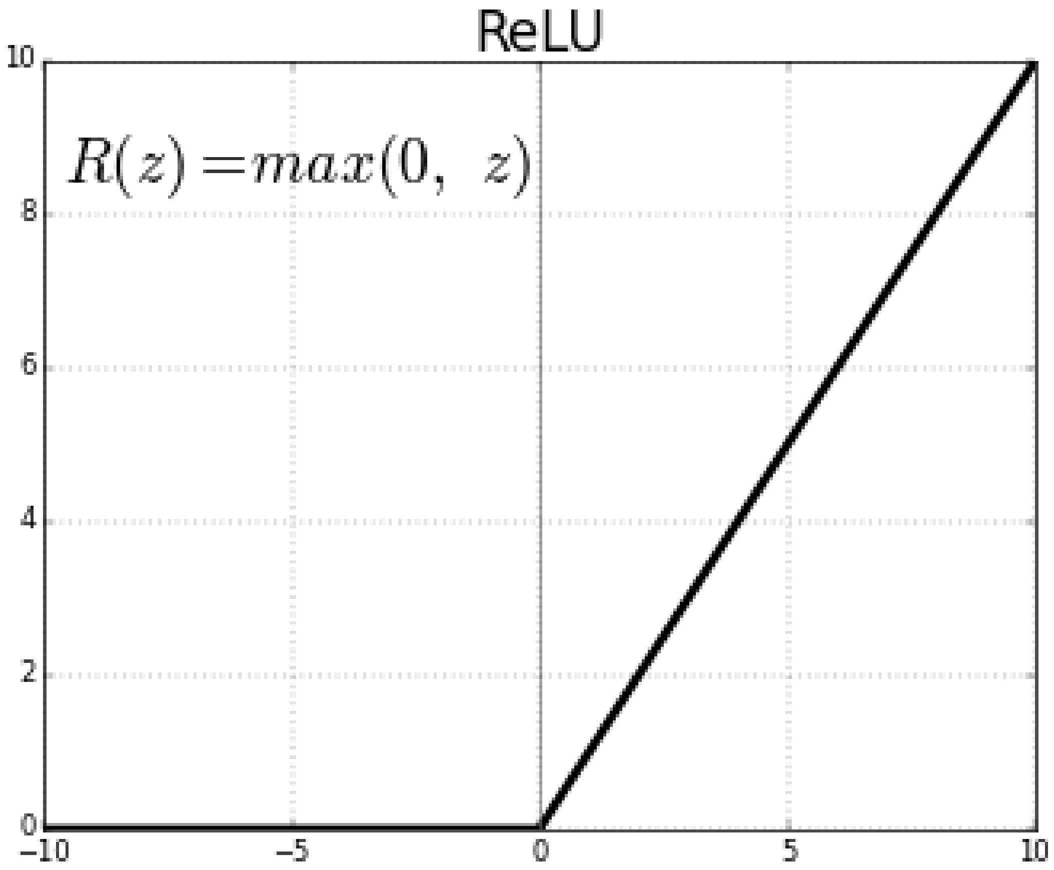


https://skymind.ai/wiki/lstm

Gradient Vanishing



Rectified Linear Unit (ReLU) Activation

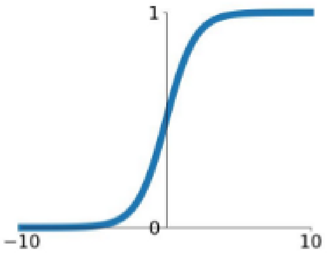


Review from Last Lecture

Activation Functions

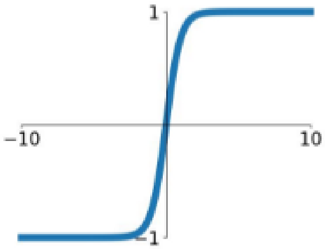
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



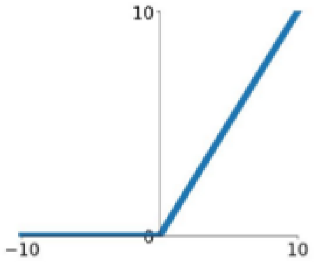
tanh

$\tanh(x)$



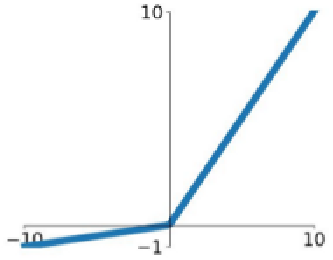
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

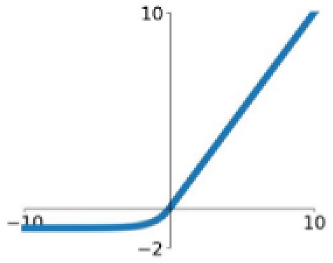


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

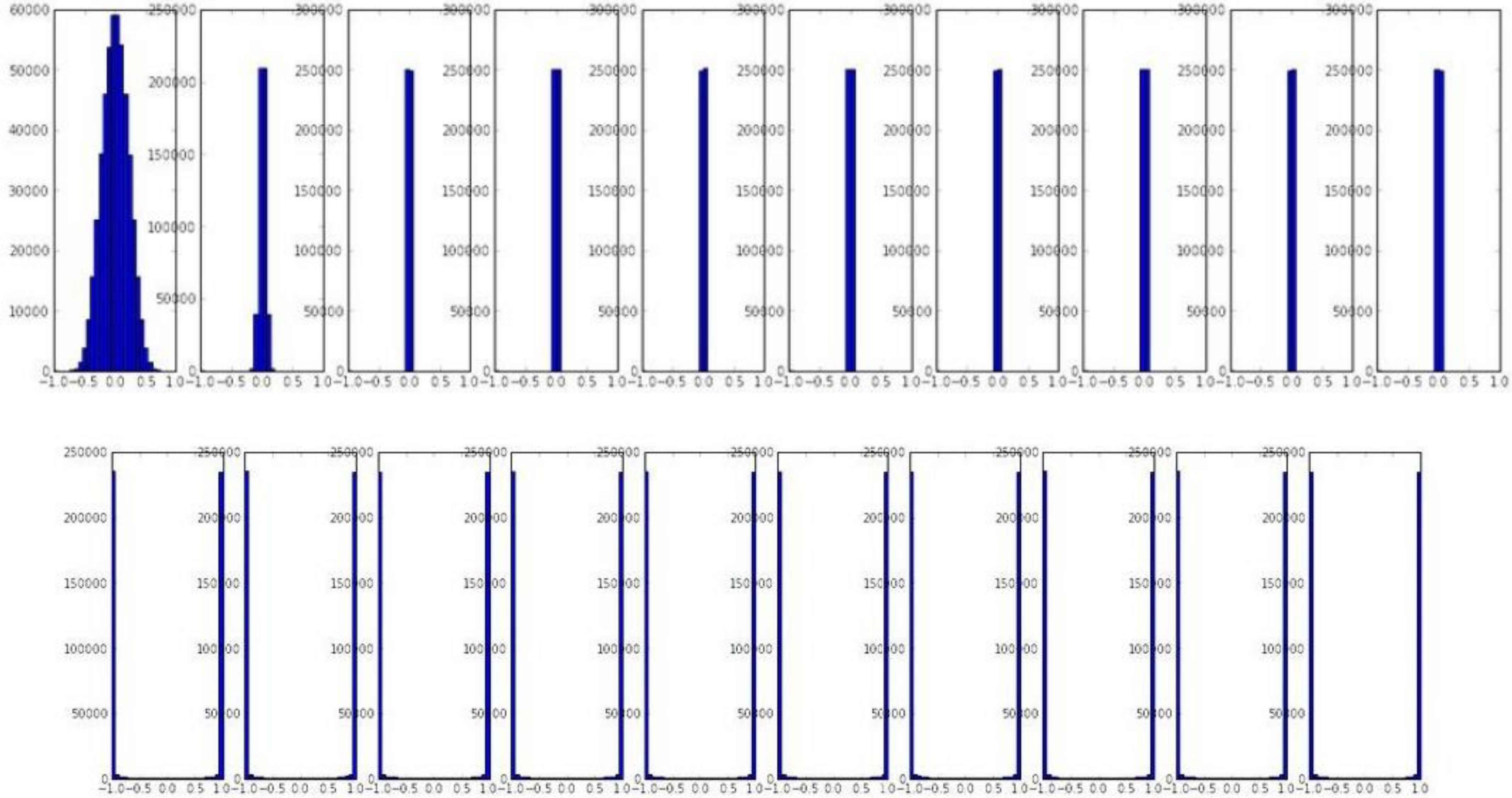
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Other Tenchniques
Xavier Initialization and Batch Normalization

Xavier Initialization

Xavier Initialization



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.