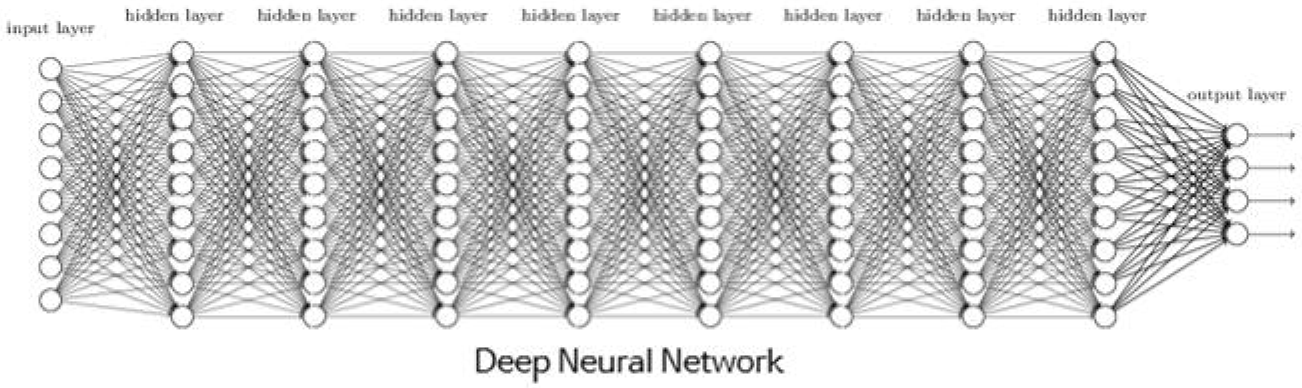
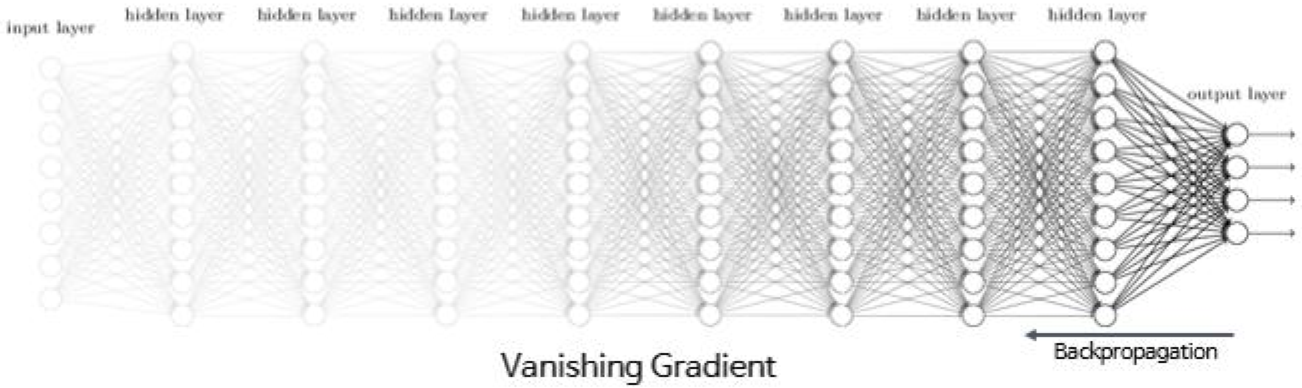
AI VIETNAM
All-in-One Course
(TA Session)

VANISHING PROBLEM





Content

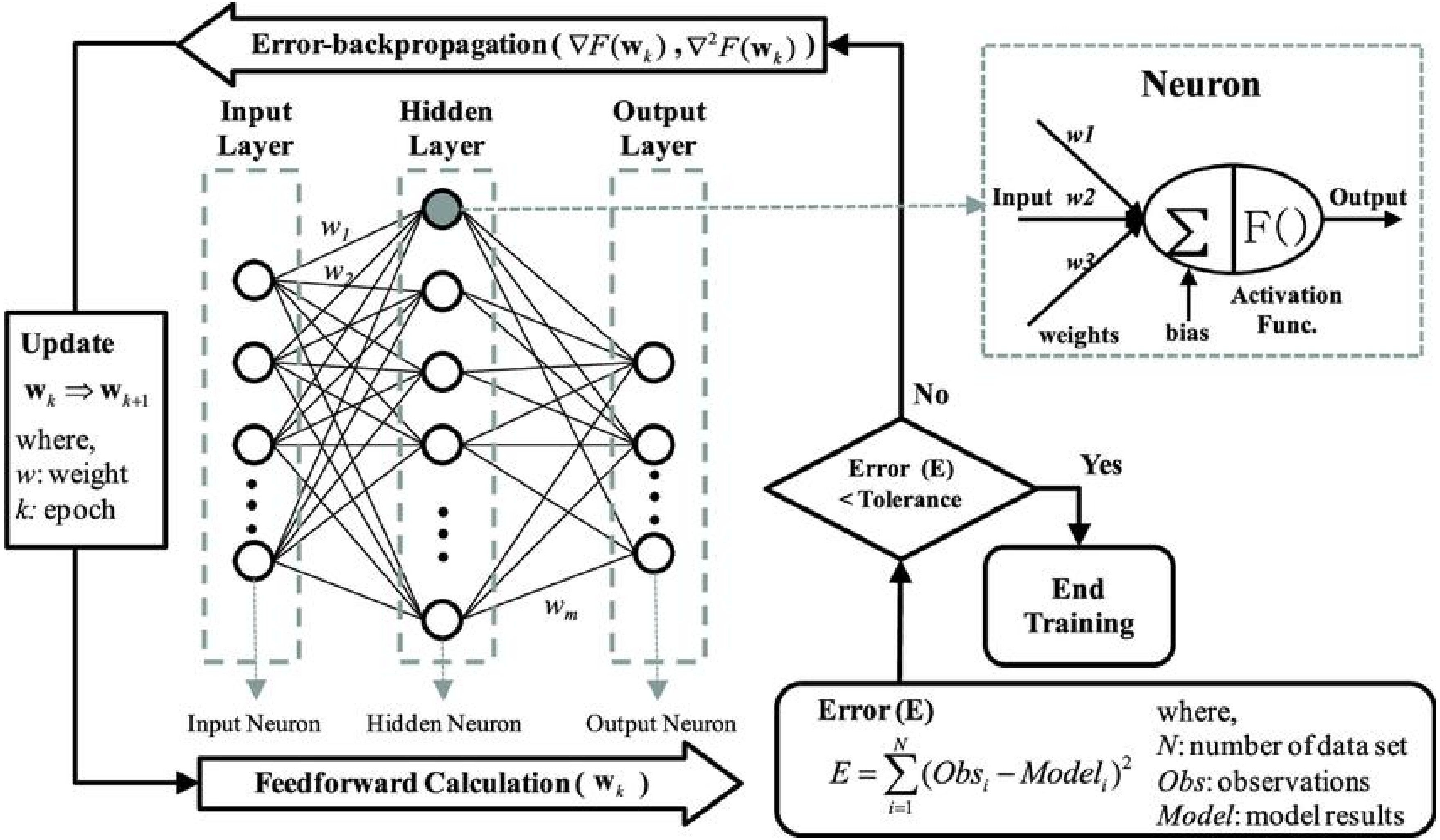
- **Giới thiệu Vanishing và Exploding Problem**
  - Vanishing Problem
  - Exploding Problem
- **Fashion MNIST Vanishing Problem**
  - Giới thiệu vấn đề
  - Solution1: Weight Increasing
  - Solution2: Better Activation
  - Solution3: Better Optimizer
  - Solution4: Normalize Inside Network
  - Solution5: Skip Connection
  - Solution6: Train Some Layer
- **Other Methods**

Content

- **Giới thiệu Vanishing và Exploding Problem**
  - Vanishing Problem
  - Exploding Problem
- **Fashion MNIST Vanishing Problem**
  - Giới thiệu vấn đề
  - Solution1: Weight Increasing
  - Solution2: Better Activation
  - Solution3: Better Optimizer
  - Solution4: Normalize Inside Network
  - Solution5: Skip Connection
  - Solution6: Train Some Layer
- Other Methods

Giới thiệu Vanishing và Exploding Problem

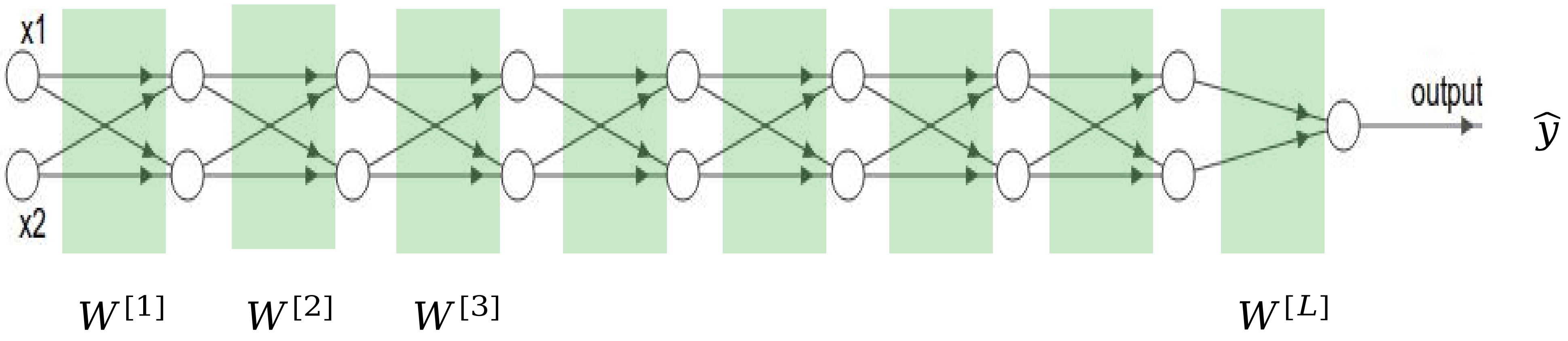
- **General Learning Circle Diagram**



https://www.analyticsvidhya.com/blog/2021/06/the-challenge-of-vanishing-exploding-gradients-in-deep-neural-networks/

Giới thiệu Vanishing và Exploding Problem

- **Forwarding**



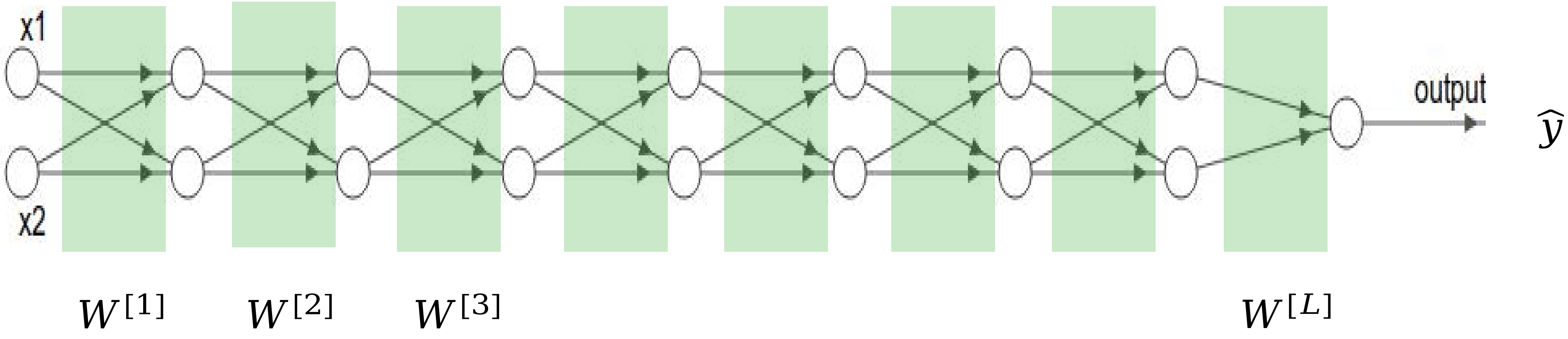
- $a^0 = x$
- $z^{[l]} = W^{[l]T} * a^{[l-1]}$
- $a^{[l]} = g(z^{[l]})$
- $\widehat{y} = a^{[L]} = g(z^{[L]}) = g(W^{[L]T} * g(W^{[L-1]T} * ... g(W^{[2]T} * g(W^{[1]T}x)))$

if $g(\bullet)$ là **linear (identity function)**

=> $\widehat{y} = a^{[L]} = g(z^{[L]}) = W^{[L]T} * W^{[L-1]T} * ... W^{[2]T} * W^{[1]T}x$

Giới thiệu Vanishing và Exploding Problem

- **Backpropagation Algorithm**



- $\frac{\partial L}{\partial w^{[L]}} = \frac{\partial L}{\partial a^{[L]}} * \frac{\partial a^{[L]}}{\partial z^{[L]}} * \frac{\partial z^{[L]}}{\partial w^{[L]}}$
- $\frac{\partial L}{\partial w^{[1]}} = \frac{\partial L}{\partial a^{[L]}} * \frac{\partial a^{[L]}}{\partial z^{[L]}} * \frac{\partial z^{[L]}}{\partial a^{[L-1]}} \dots * \frac{\partial a^{[2]}}{\partial z^{[2]}} * \frac{\partial z^{[2]}}{\partial a^{[1]}} * \frac{\partial a^{[1]}}{\partial z^{[1]}} * \frac{\partial z^{[1]}}{\partial w^{[1]}}$

if $g(\bullet)$ là **linear (identity function)**

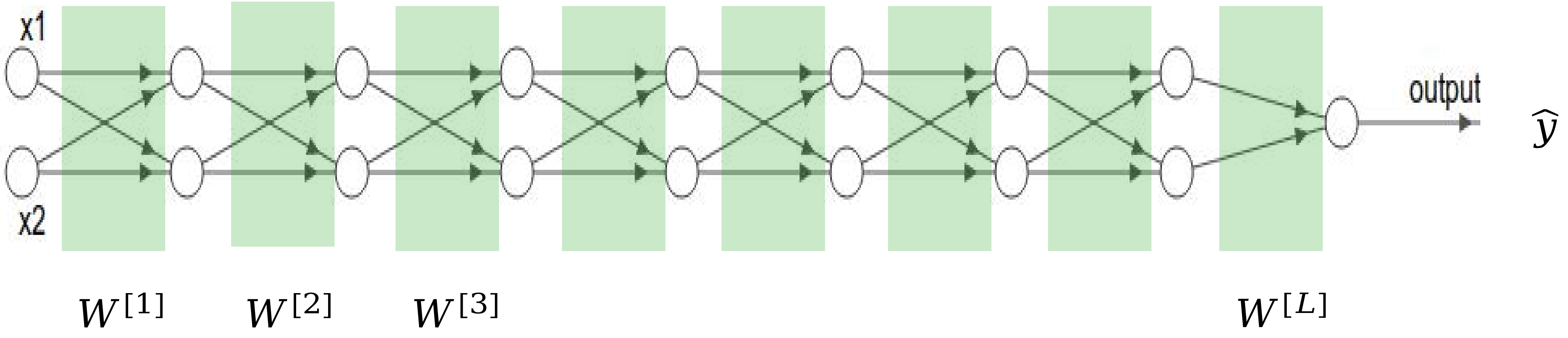
- $\frac{\partial L}{\partial w^{[1]}} = \frac{\partial L}{\partial a^{[L]}} * \frac{\partial a^{[L]}}{\partial a^{[L-1]}} * \frac{\partial a^{[L-1]}}{\partial a^{[L-2]}} * \dots * \frac{\partial a^{[2]}}{\partial a^{[1]}} * \frac{\partial a^{[1]}}{\partial w^{[1]}}$

- $Loss = L(\widehat{y}, y),\ \ \widehat{y} = a^{[L]}$
- $a^{[l]} = g(z^{[l]})$

$$\boldsymbol{w^{[l]} = w^{[l]} - \eta \frac{\partial L}{\partial w^{l}}}$$

Giới thiệu Vanishing và Exploding Problem

- **Exploding Problem**



- $a^0 = x$
- $z^{[l]} = W^{[l]T} * a^{[l-1]}$
- $a^{[l]} = g(z^{[l]})$
- $\widehat{y} = a^{[L]} = g(z^{[L]}) = g(W^{[L]T} * g(W^{[L-1]T} * \ldots g(W^{[2]T} * g(W^{[1]T} x)))$

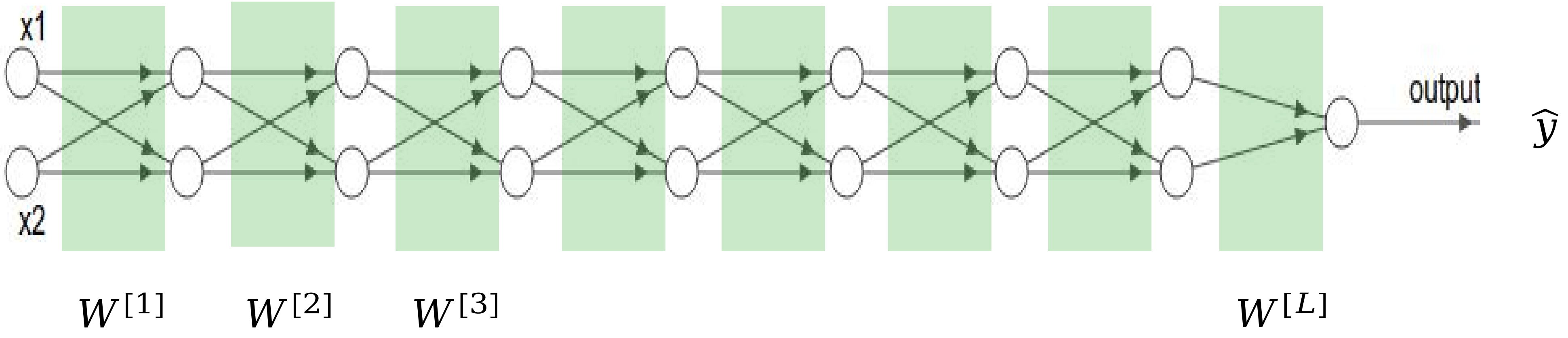
if $g(\bullet)$ là **linear (identity function)**

=> $\widehat{y} = a^{[L]} = g(z^{[L]}) = W^{[L]T} * W^{[L-1]T} * \ldots W^{[2]T} * W^{[1]T} x$

Tất cả = 10 **=> chỉ 7 layers $\widehat{\mathbf{y}} = \mathbf{10^7}$**

Giới thiệu Vanishing và Exploding Problem

- **Exploding Problem**



Tất cả = 10 => chỉ 7 layers $= 10^7$

- $\dfrac{\partial L}{\partial w^{[L]}} = \dfrac{\partial L}{\partial a^{[L]}} * \dfrac{\partial a^{[L]}}{\partial z^{[L]}} * \dfrac{\partial z^{[L]}}{\partial w^{[L]}}$
- $\dfrac{\partial L}{\partial w^{[1]}} = \dfrac{\partial L}{\partial a^{[L]}} * \dfrac{\partial a^{[L]}}{\partial z^{[L]}} * \dfrac{\partial z^{[L]}}{\partial a^{[L-1]}} \dots * \dfrac{\partial a^{[2]}}{\partial z^{[2]}} * \dfrac{\partial z^{[2]}}{\partial a^{[1]}} * \dfrac{\partial a^{[1]}}{\partial z^{[1]}} * \dfrac{\partial z^{[1]}}{\partial w^{[1]}}$

if $g(\bullet)$ là **linear (identity function)**

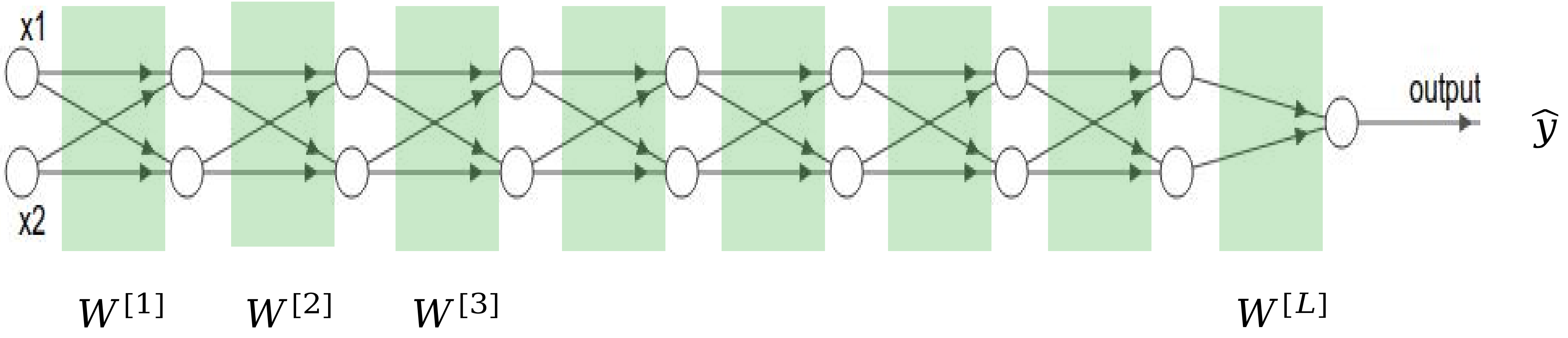
- $\dfrac{\partial L}{\partial w^{[1]}} = \dfrac{\partial L}{\partial a^{[L]}} * \dfrac{\partial a^{[L]}}{\partial a^{[L-1]}} * \dfrac{\partial a^{[L-1]}}{\partial a^{[L-2]}} * \dots * \dfrac{\partial a^{[2]}}{\partial a^{[1]}} * \dfrac{\partial a^{[1]}}{\partial w^{[1]}}$

- $Loss = L(\hat{y}, y),\ \hat{y} = a^{[L]}$
- $a^{[l]} = g(z^{[l]})$

$$\boldsymbol{w^{[l]} = w^{[l]} - \eta \frac{\partial L}{\partial w^{l}}}$$

Giới thiệu Vanishing và Exploding Problem

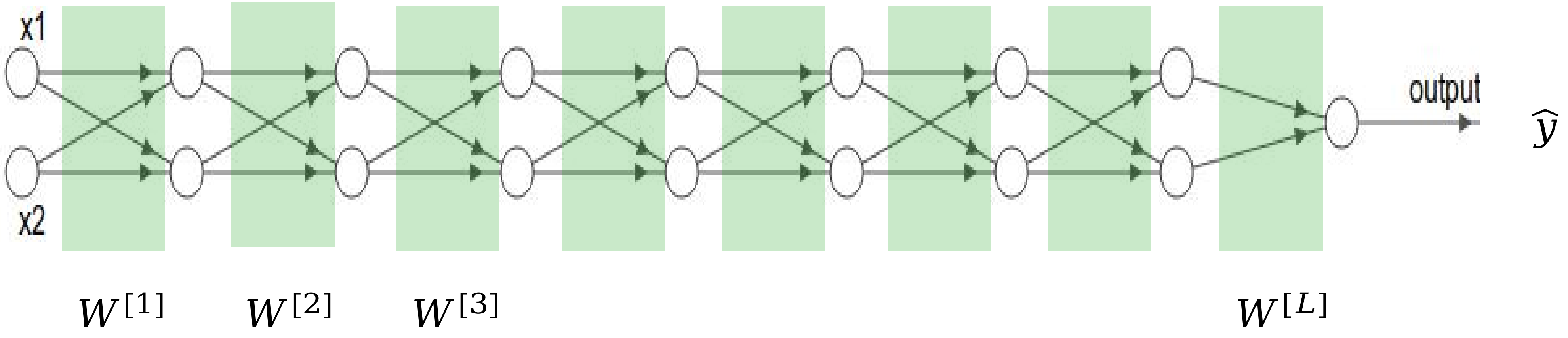
- **Exploding Problem**



- *"exploding gradients can make learning unstable".* Page 282, Deep Learning (by Goodfellow, Yoshua Bengio, Aaron Courville), 2016
- **Exploding** trong **trường hợp tốt nhất** việc update weight một lượng lớn làm **network** học **không ổn định** và **không thể hội tụ**
- **Exploding** trong **trường hợp xấu nhất NaN weight không thể update**
- Một số dấu hiệu của exploding: loss là NaN ,loss rất lớn và không có dấu hiệu giảm, model không ổn định loss tăng giảm không ổn định nhưng nhìn chung vẫn lớn

Giới thiệu Vanishing và Exploding Problem

- **Vanishing Problem**



- $a^0 = x$
- $z^{[l]} = W^{[l]T} * a^{[l-1]}$
- $a^{[l]} = g(z^{[l]})$
- $\widehat{y} = a^{[L]} = g(z^{[L]}) = g(W^{[L]T} * g(W^{[L-1]T} * \ldots g(W^{[2]T} * g(W^{[1]T} x)))$

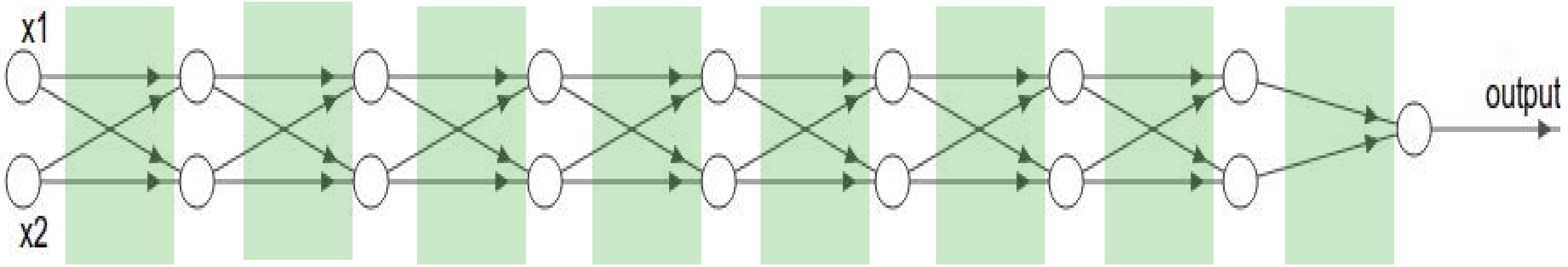
if $g(\bullet)$ là **linear (identity function)**

=> $\widehat{y} = a^{[L]} = g(z^{[L]}) = W^{[L]T} * W^{[L-1]T} * \ldots W^{[2]T} * W^{[1]T} x$

Tất cả = 0.1 **=> chỉ 7 layers $\widehat{y} = 0.1^7$**

Giới thiệu Vanishing và Exploding Problem

- **Vanishing Problem**



$\widehat{y}$

$W^{[1]}$ $W^{[2]}$ $W^{[3]}$ $W^{[L]}$

Tất cả = 0.1 **=> chỉ 7 layers = 0.1^7**

- $\frac{\partial L}{\partial w^{[L]}} = \frac{\partial L}{\partial a^{[L]}} * \frac{\partial a^{[L]}}{\partial z^{[L]}} * \frac{\partial z^{[L]}}{\partial w^{[L]}}$
- $\frac{\partial L}{\partial w^{[1]}} = \frac{\partial L}{\partial a^{[L]}} * \frac{\partial a^{[L]}}{\partial z^{[L]}} * \frac{\partial z^{[L]}}{\partial a^{[L-1]}} \cdots * \frac{\partial a^{[2]}}{\partial z^{[2]}} * \frac{\partial z^{[2]}}{\partial a^{[1]}} * \frac{\partial a^{[1]}}{\partial z^{[1]}} * \frac{\partial z^{[1]}}{\partial w^{[1]}}$

if $g(\bullet)$ là **linear (identity function)**

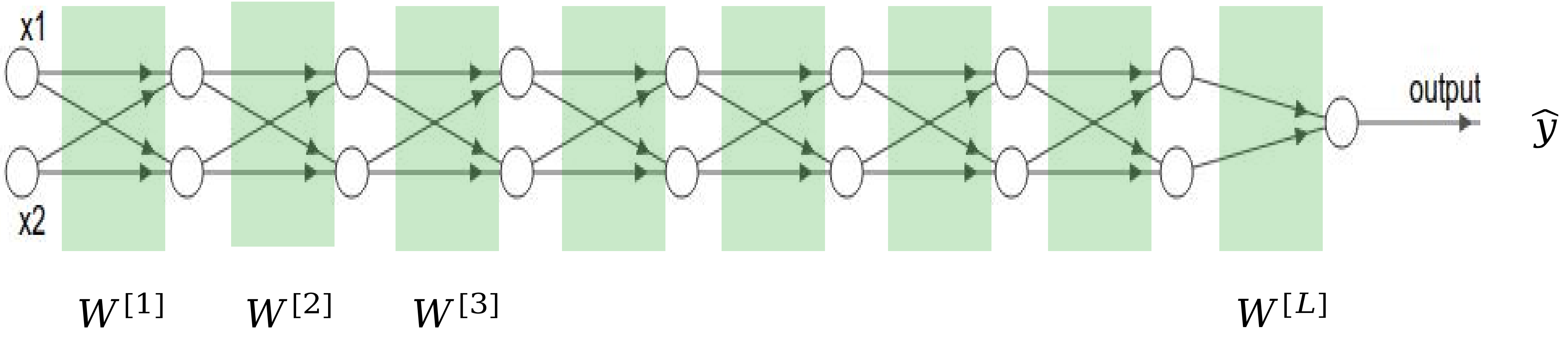
- $\frac{\partial L}{\partial w^{[1]}} = \frac{\partial L}{\partial a^{[L]}} * \frac{\partial a^{[L]}}{\partial a^{[L-1]}} * \frac{\partial a^{[L-1]}}{\partial a^{[L-2]}} * \cdots * \frac{\partial a^{[2]}}{\partial a^{[1]}} * \frac{\partial a^{[1]}}{\partial w^{[1]}}$

- $Loss = L(\widehat{y}, y),\ \widehat{y} = a^{[L]}$
- $a^{[l]} = g(z^{[l]})$

$$\boldsymbol{w^{[l]} = w^{[l]} - \eta \frac{\partial L}{\partial w^{l}}}$$

Giới thiệu Vanishing và Exploding Problem

- **Vanishing Problem**



- **Backpropagation** dùng **chain rule**, khi tính **loss** sẽ là **tích của các gradient** trong **từng layer**
- Gradient càng nhỏ khi nhân lại với nhau sẽ càng tiến về 0
- Parameter ở các **layer gần input** sẽ **không đóng góp** vào việc học của model

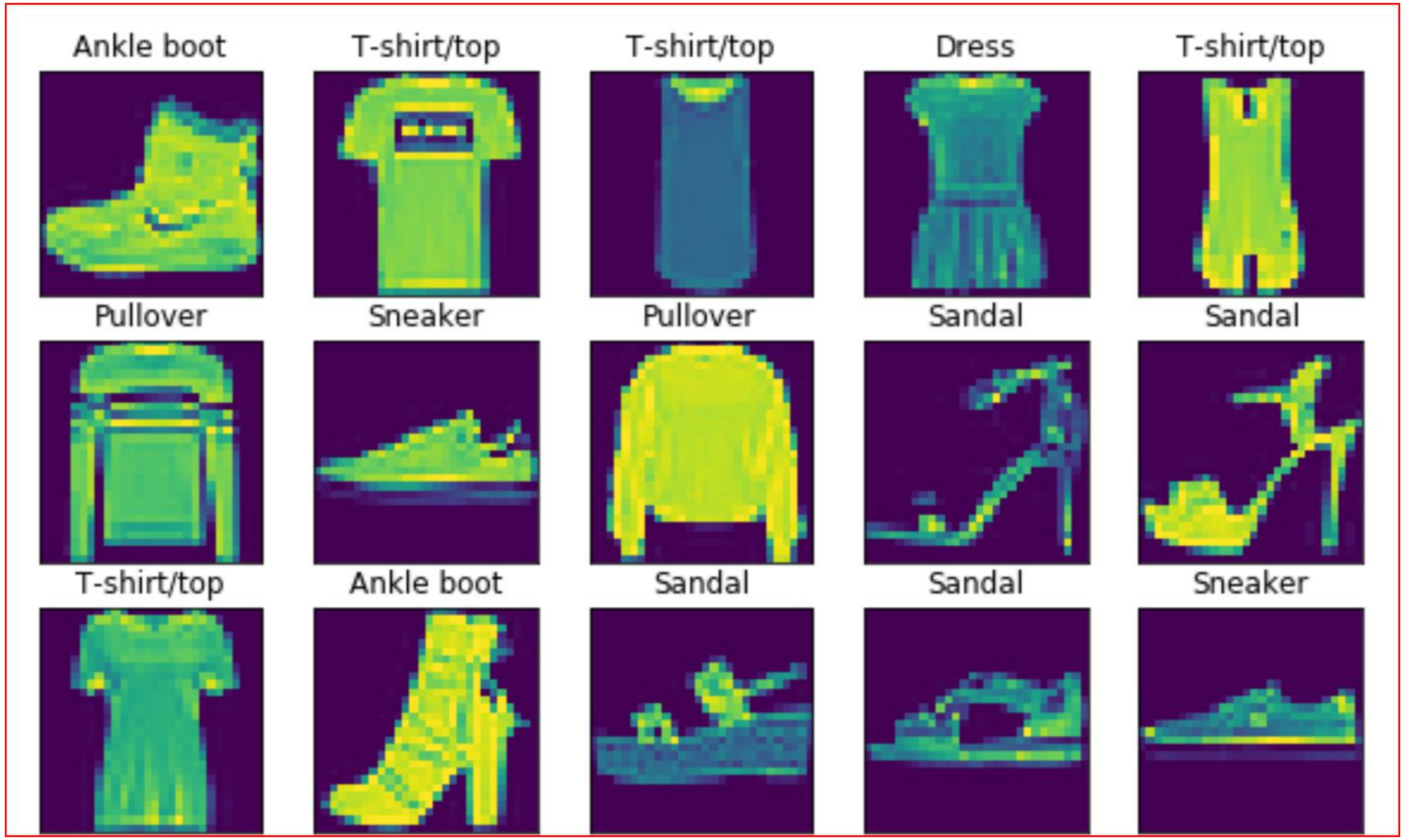
Giới thiệu Vanishing và Exploding Problem

Content

- **Giới thiệu Vanishing và Exploding Problem**
  - Vanishing Problem
  - Exploding Problem
- **Fashion MNIST Vanishing Problem**
  - Giới thiệu vấn đề
  - Solution1: Weight Increasing
  - Solution2: Better Activation
  - Solution3: Better Optimizer
  - Solution4: Normalize Inside Network
  - Solution5: Skip Connection
  - Solution6: Train Some Layer

Fashion MNIST Vanishing Problem

- **Giới thiệu vấn đề**
  - Fashion MNIS dataset
    - **Train**: 60,000 samples
    - **Test**: 10,000 samples
    - **Classes**: 10
    - **Size**: 28x28
    - **Image type**: grayscale

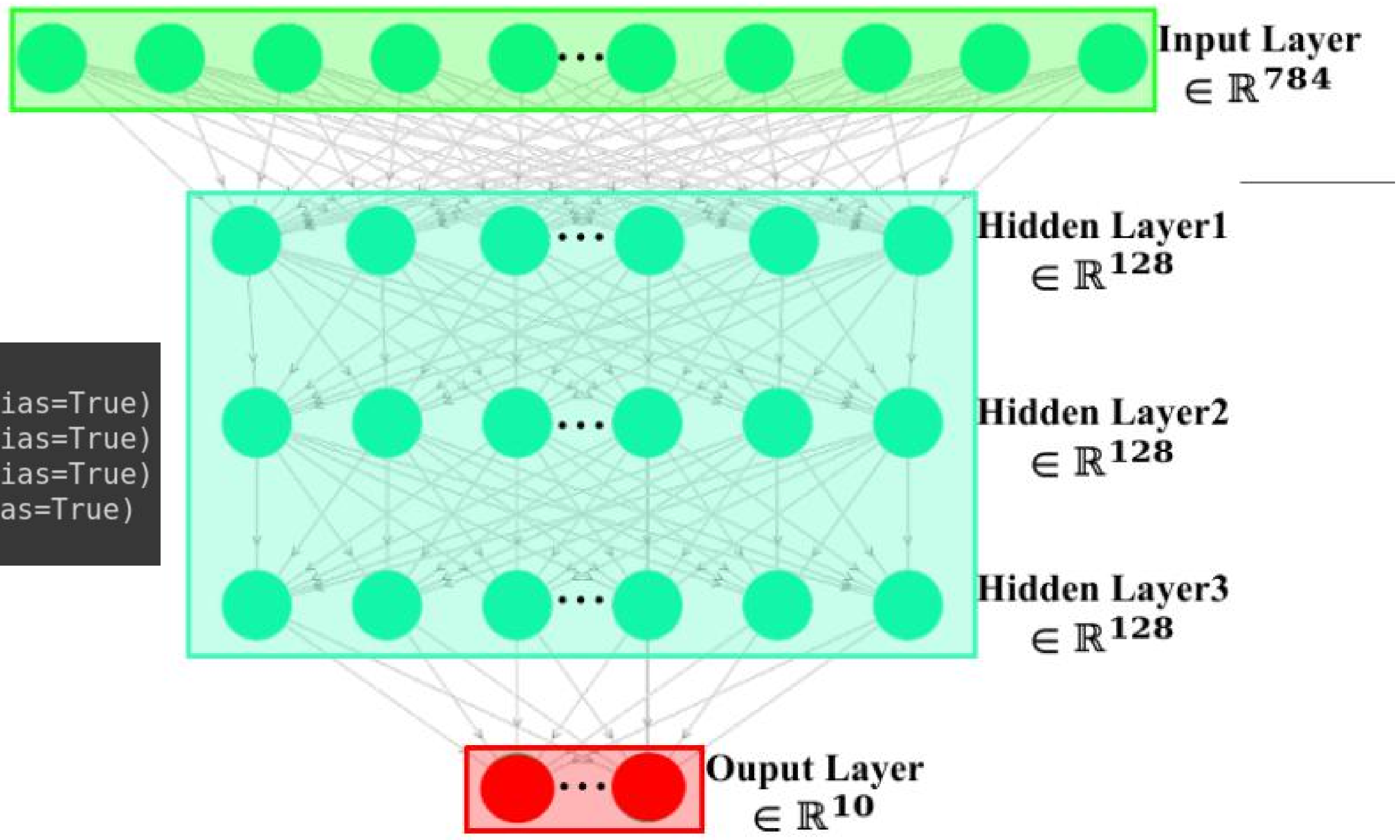


https://github.com/zalandoresearch/fashion-mnist
https://miro.medium.com/max/1838/1*6YhvuUHE0LPHEsqU_Cis9w.png

Fashion MNIST Vanishing Problem

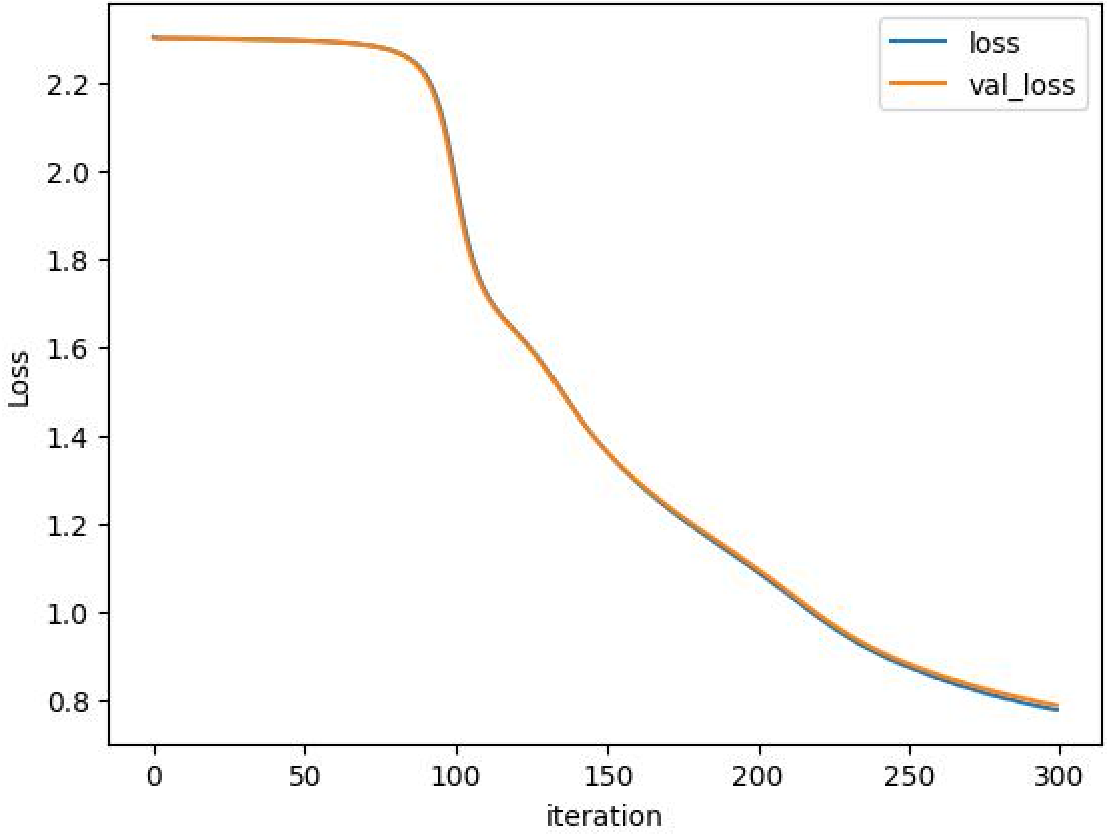
- **Giới thiệu vấn đề**
  - Model1:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: 3 layers
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: CE
    - **Optimizer**: sgd

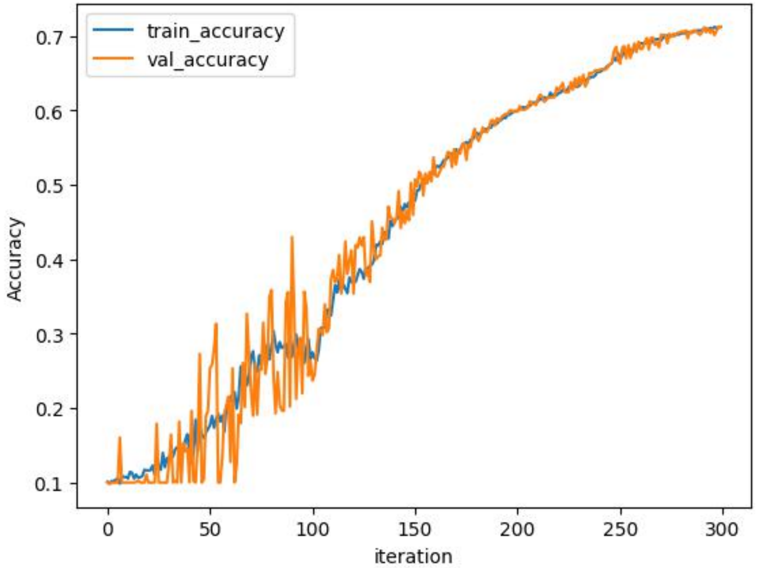
```
MLP(
  (layer1): Linear(in_features=784, out_features=128, bias=True)
  (layer2): Linear(in_features=128, out_features=128, bias=True)
  (layer3): Linear(in_features=128, out_features=128, bias=True)
  (output): Linear(in_features=128, out_features=10, bias=True)
)
```



Fashion MNIST Vanishing Problem

- **Giới thiệu vấn đề**
  - Model1:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: 3 layers
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: CE
    - **Optimizer**: sgd

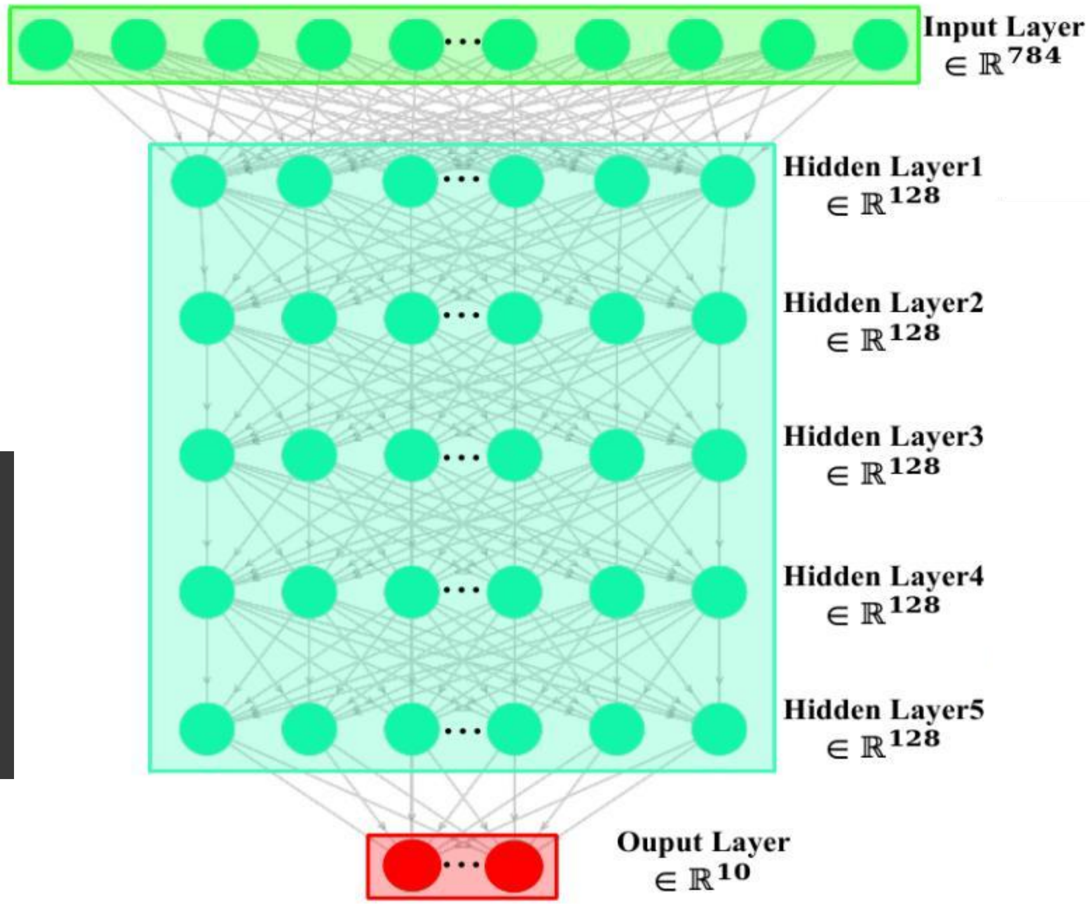




Fashion MNIST Vanishing Problem

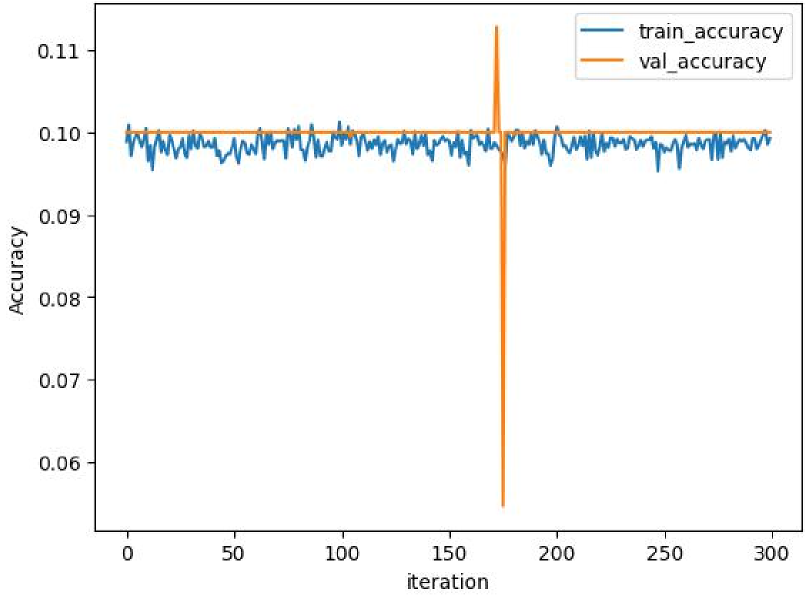
- **Giới thiệu vấn đề**
  - Model2:
    - **Weight Initialization**: $\mu=0$, $\sigma=0.05$
    - **Hidden Layers**: 5 layers
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: CE
    - **Optimizer**: sgd

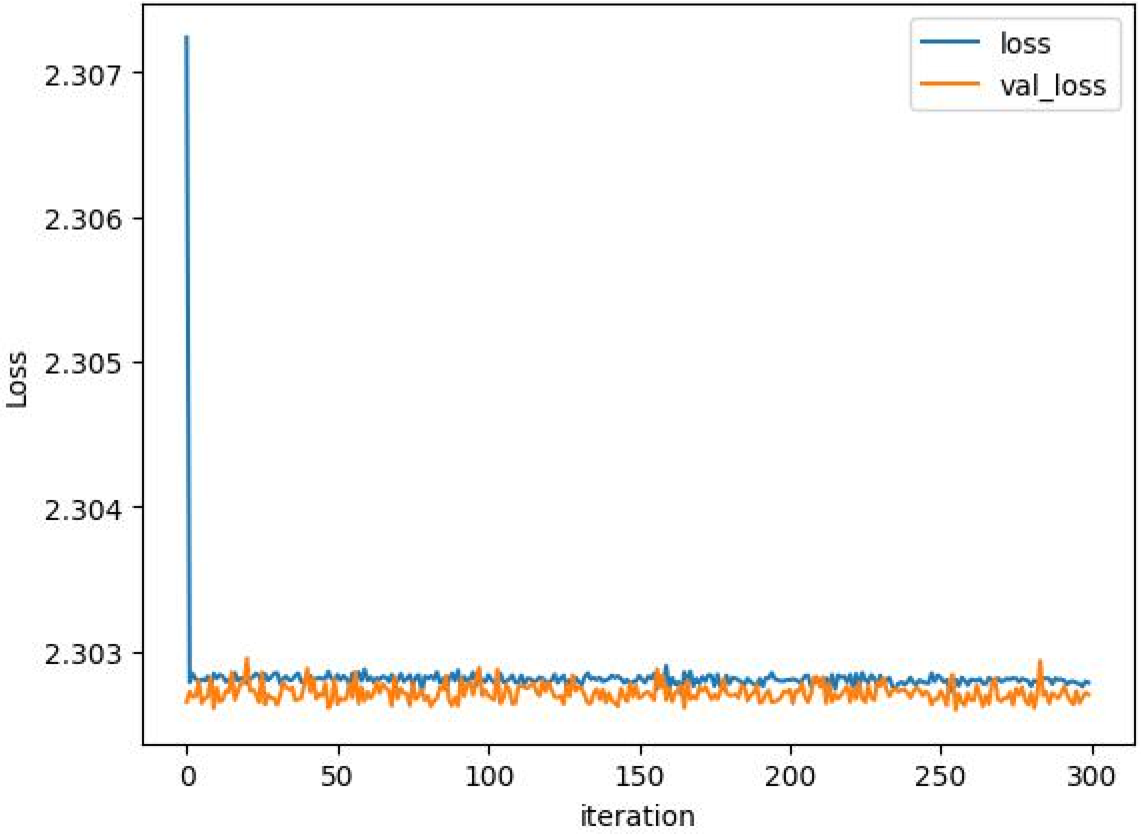
```
MLP(
  (layer1): Linear(in_features=784, out_features=128, bias=True)
  (layer2): Linear(in_features=128, out_features=128, bias=True)
  (layer3): Linear(in_features=128, out_features=128, bias=True)
  (layer4): Linear(in_features=128, out_features=128, bias=True)
  (layer5): Linear(in_features=128, out_features=128, bias=True)
  (output): Linear(in_features=128, out_features=10, bias=True)
)
```



Fashion MNIST Vanishing Problem

- **Giới thiệu vấn đề**
  - Model2:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: 5 layers
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: CE
    - **Optimizer**: sgd

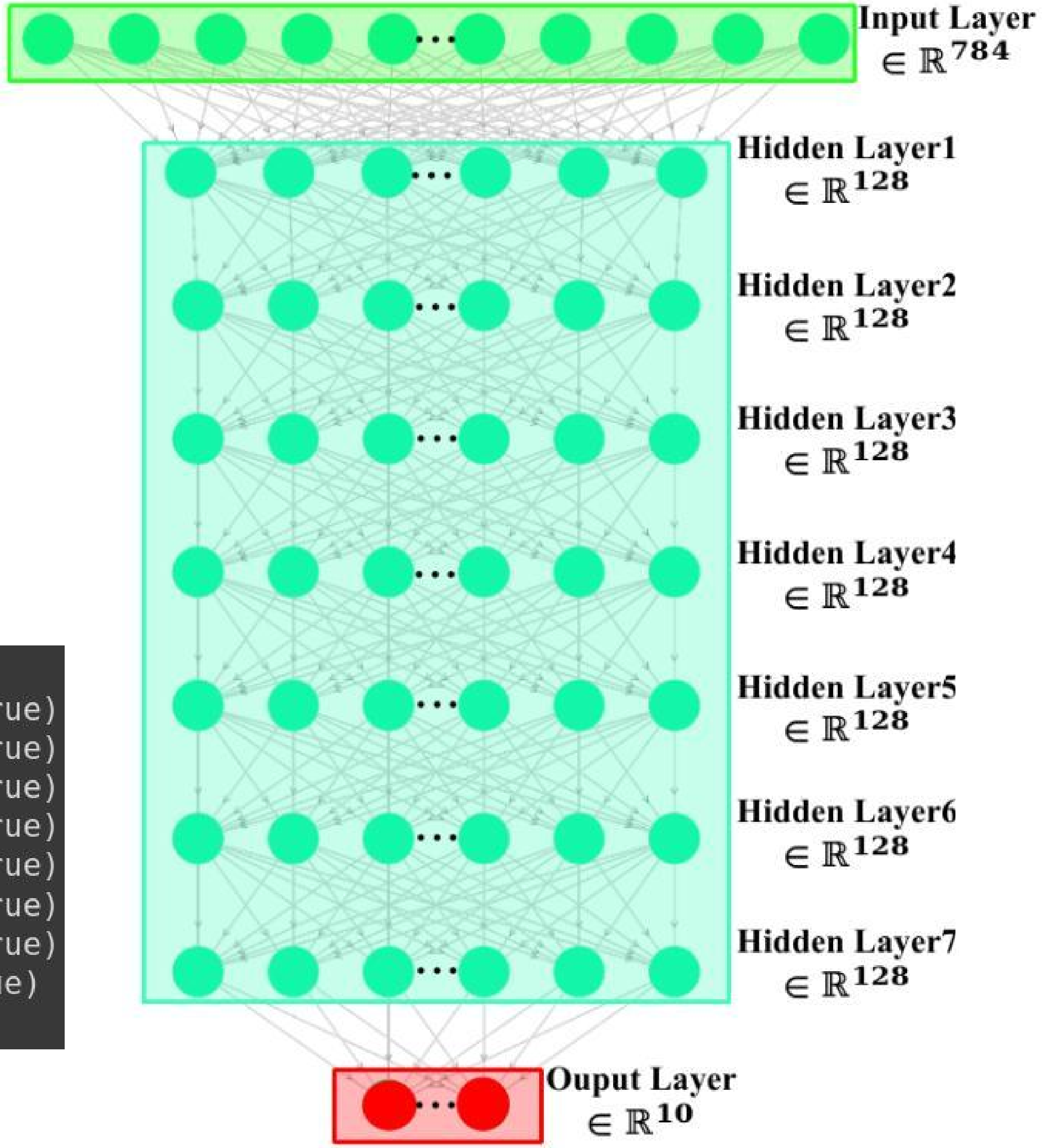




Fashion MNIST Vanishing Problem

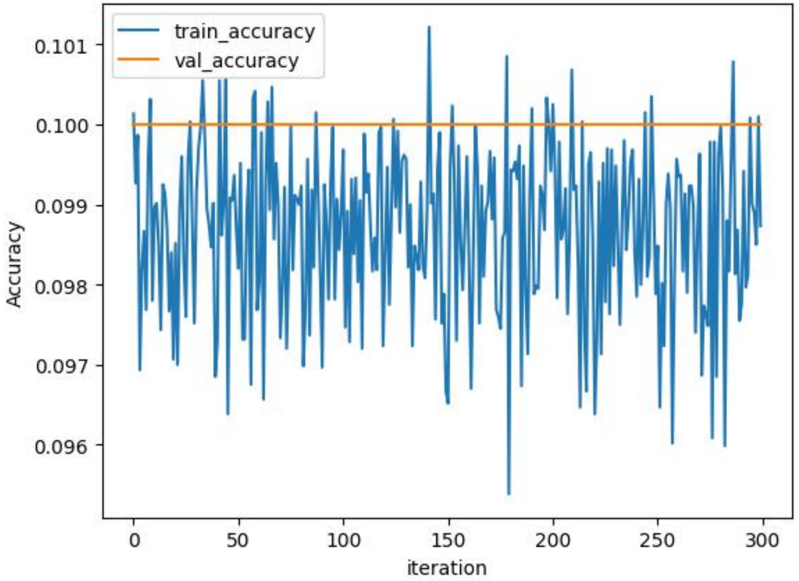
- **Giới thiệu vấn đề**
  - Model3:
    - **Weight Initialization**: $\mu=0$, $\sigma=0.05$
    - **Hidden Layers**: 7 layers
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: CE
    - **Optimizer**: sgd

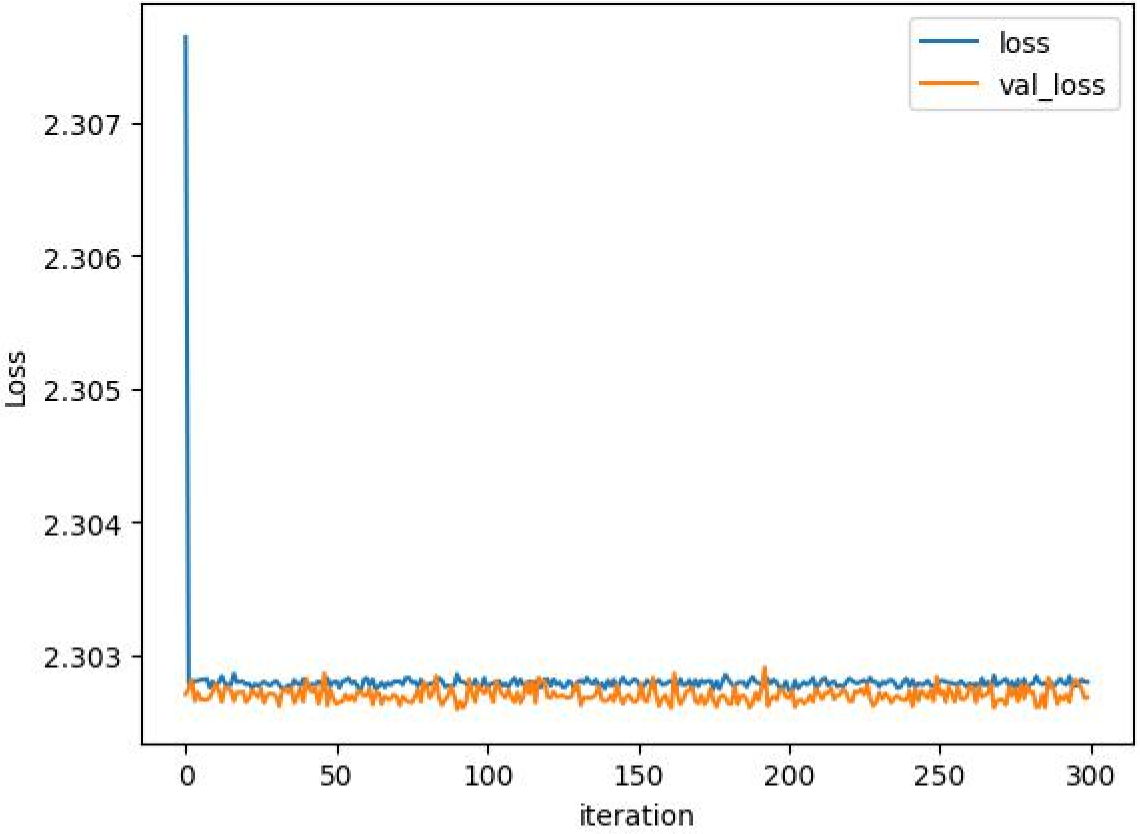
```
MLP(
  (layer1): Linear(in_features=784, out_features=128, bias=True)
  (layer2): Linear(in_features=128, out_features=128, bias=True)
  (layer3): Linear(in_features=128, out_features=128, bias=True)
  (layer4): Linear(in_features=128, out_features=128, bias=True)
  (layer5): Linear(in_features=128, out_features=128, bias=True)
  (layer6): Linear(in_features=128, out_features=128, bias=True)
  (layer7): Linear(in_features=128, out_features=128, bias=True)
  (output): Linear(in_features=128, out_features=10, bias=True)
)
```



Fashion MNIST Vanishing Problem

- **Giới thiệu vấn đề**
  - Model3:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: 7 layers
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: CE
    - **Optimizer**: sgd





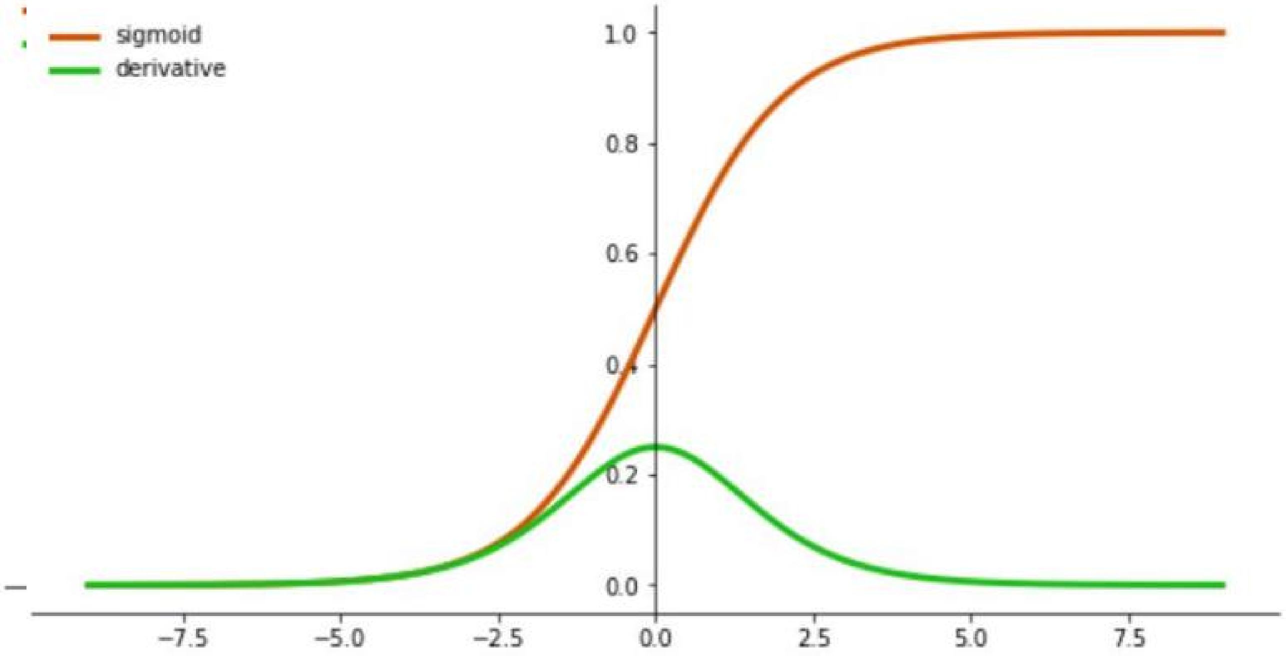
Fashion MNIST Vanishing Problem

- **Giới thiệu vấn đề**
  - Các nguyên nhân có thể gây ra vanishing problem

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

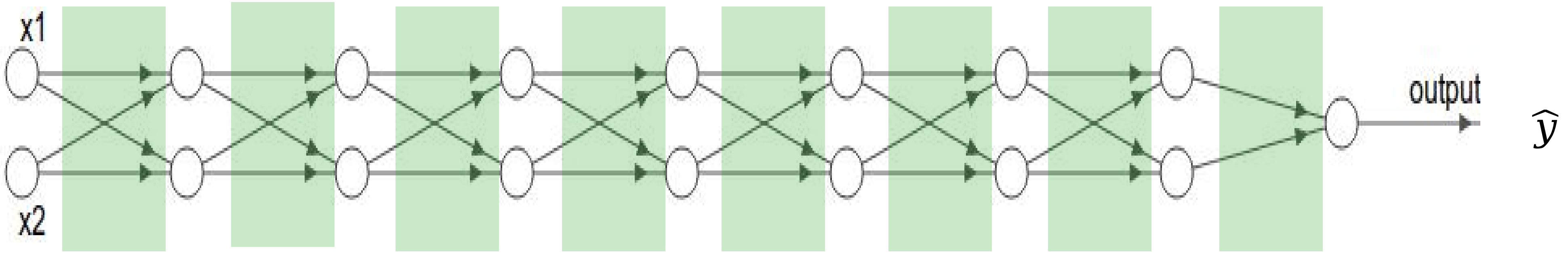
$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

Giá trị của đạo hàm:
- min = 0
- max = 0.25



Giới thiệu Vanishing và Exploding Problem

- **Giới thiệu vấn đề**
  - Các nguyên nhân có thể gây ra vanishing problem



- $\dfrac{\partial L}{\partial w^{[L]}} = \dfrac{\partial L}{\partial a^{[L]}} * \dfrac{\partial a^{[L]}}{\partial z^{[L]}} * \dfrac{\partial z^{[L]}}{\partial w^{[L]}}$

- $Loss = L(\hat{y}, y),\ \ \hat{y} = a^{[L]}$
- $a^{[l]} = g(z^{[l]})$

=> chỉ 7 layers với giá trị đạo hàm đạt tối đa cho mỗi layer = $\mathbf{0,25^7 \approx 6.1^{-5}}$

if $g(\bullet)$ là **sigmoid function** $\sigma(x)$

- $\dfrac{\partial L}{\partial w^{[1]}} = \dfrac{\partial L}{\partial a^{[L]}} * \underbrace{\dfrac{\partial a^{[L]}}{\partial z^{[L]}}}_{[0;\,0,25]} * \dfrac{\partial z^{[L]}}{\partial a^{[L-1]}} \cdots * \underbrace{\dfrac{\partial a^{[2]}}{\partial z^{[2]}}}_{[0;\,0,25]} * \dfrac{\partial z^{[2]}}{\partial a^{[1]}} * \underbrace{\dfrac{\partial a^{[1]}}{\partial z^{[1]}}}_{[0;\,0,25]} * \dfrac{\partial z^{[1]}}{\partial w^{[1]}}$

$$\boldsymbol{w^{[l]} = w^{[l]} - \eta \frac{\partial L}{\partial w^{l}}}$$

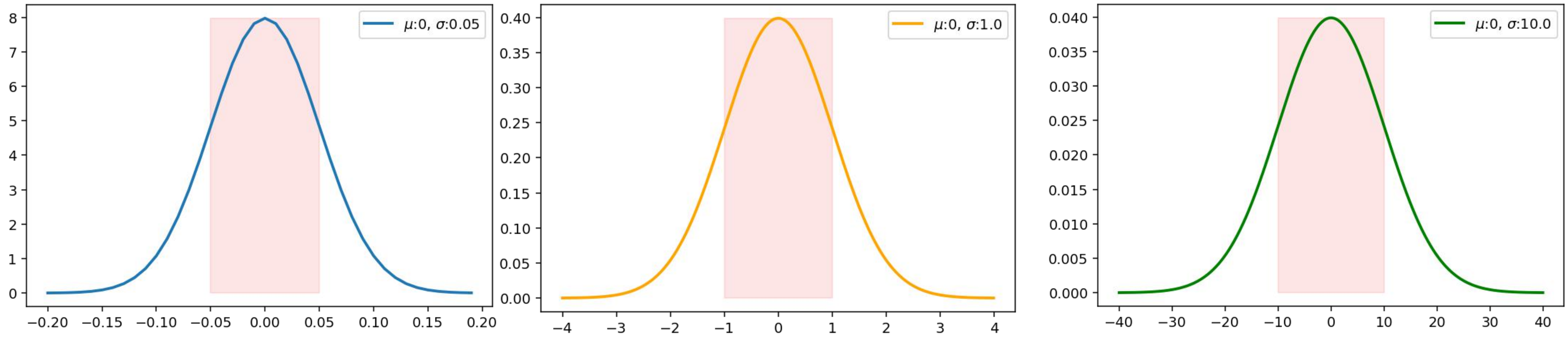
Fashion MNIST Vanishing Problem

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$f(x)$ = probability density function

σ = standard deviation

μ = mean



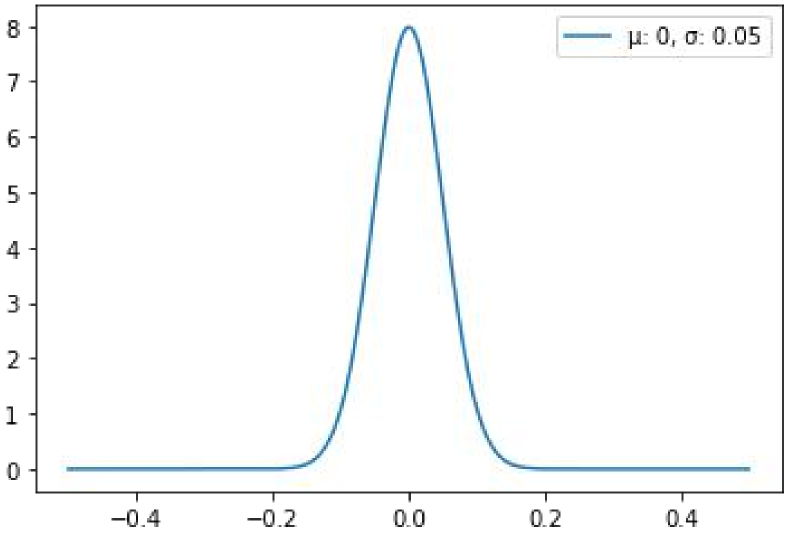
Fashion MNIST Vanishing Problem

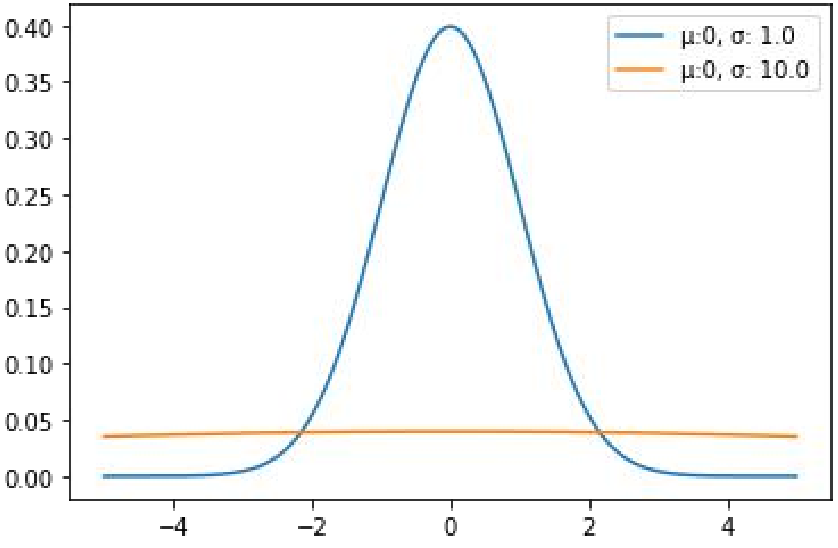
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

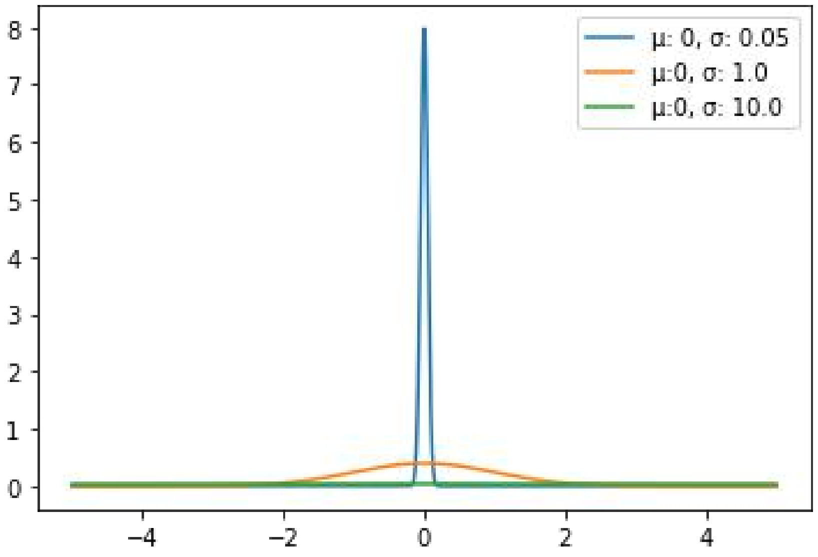
$f(x)$ = probability density function

σ = standard deviation

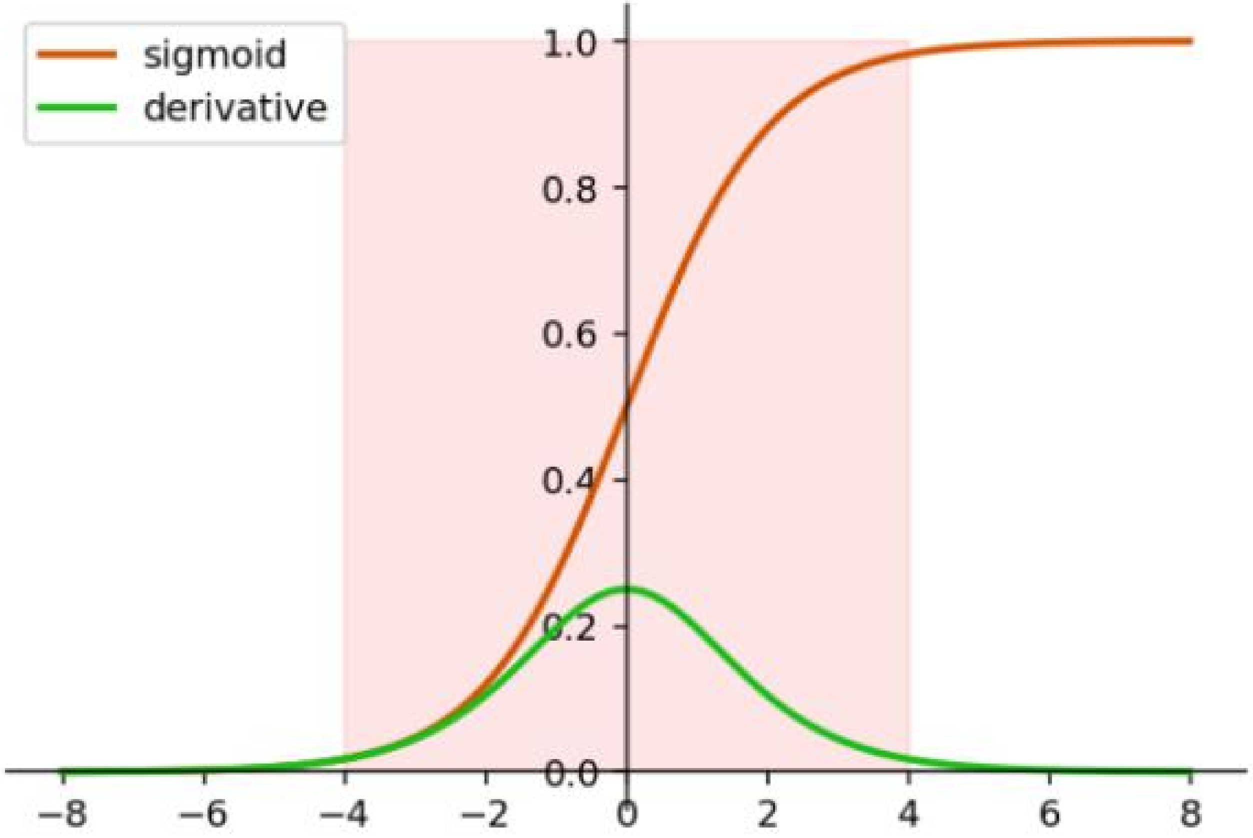
μ = mean





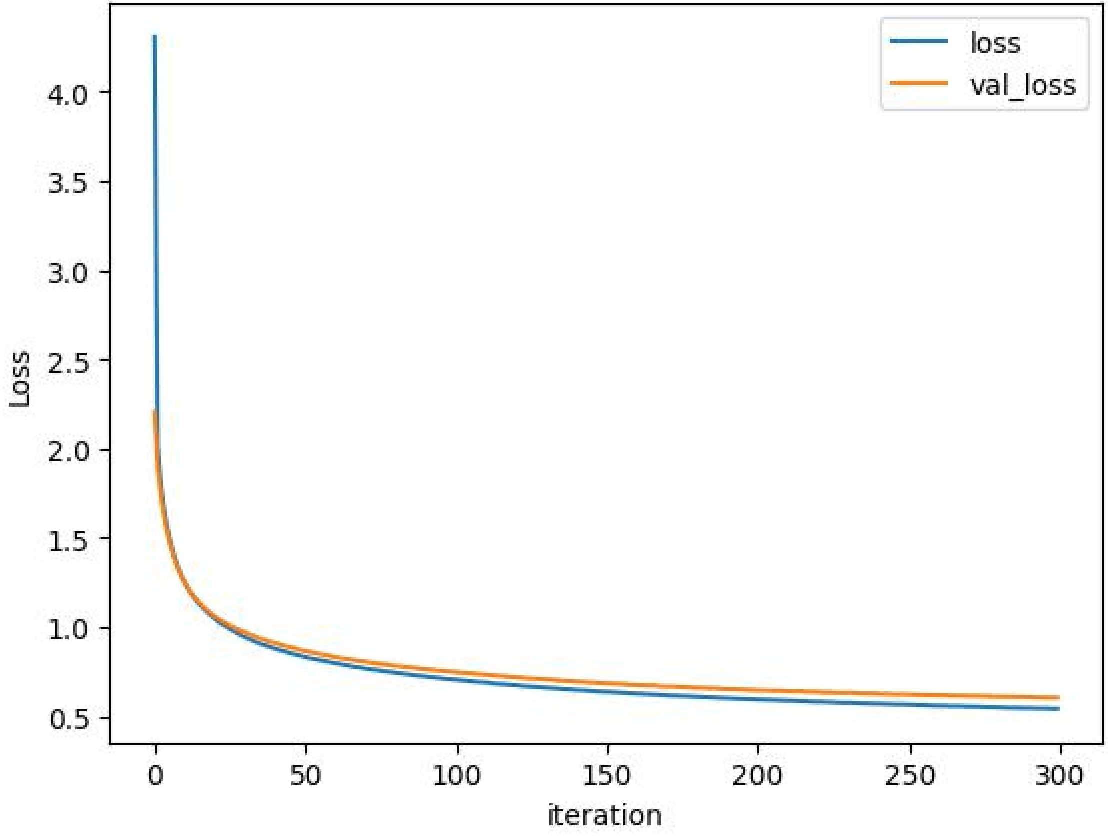


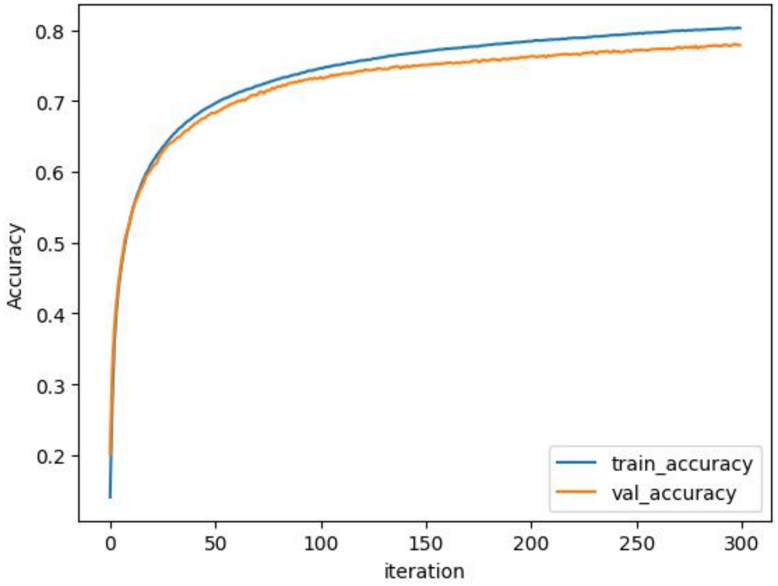
Fashion MNIST Vanishing Problem



Fashion MNIST Vanishing Problem

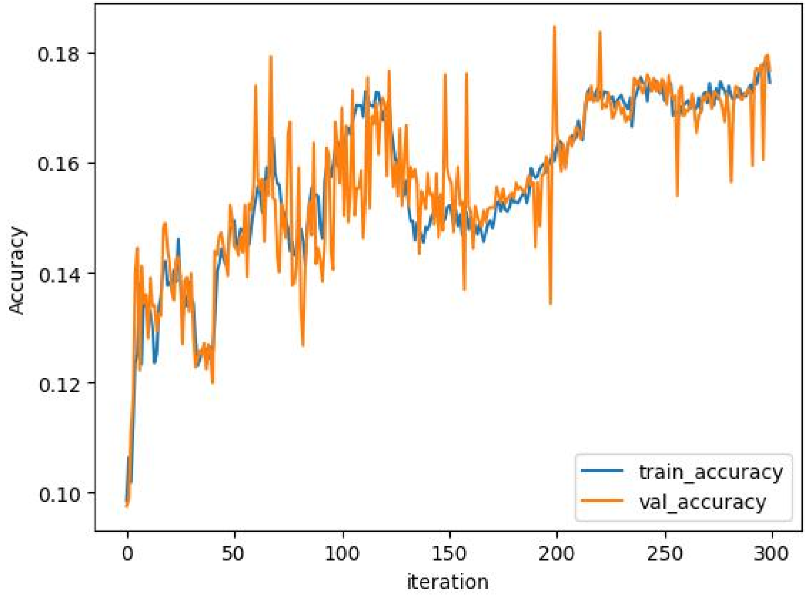
- **Weight Increasing**
  - Model:
    - **Weight Initialization**: μ=0, σ=1.0
    - **Hidden Layers**: 7 layers
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: CE
    - **Optimizer**: sgd

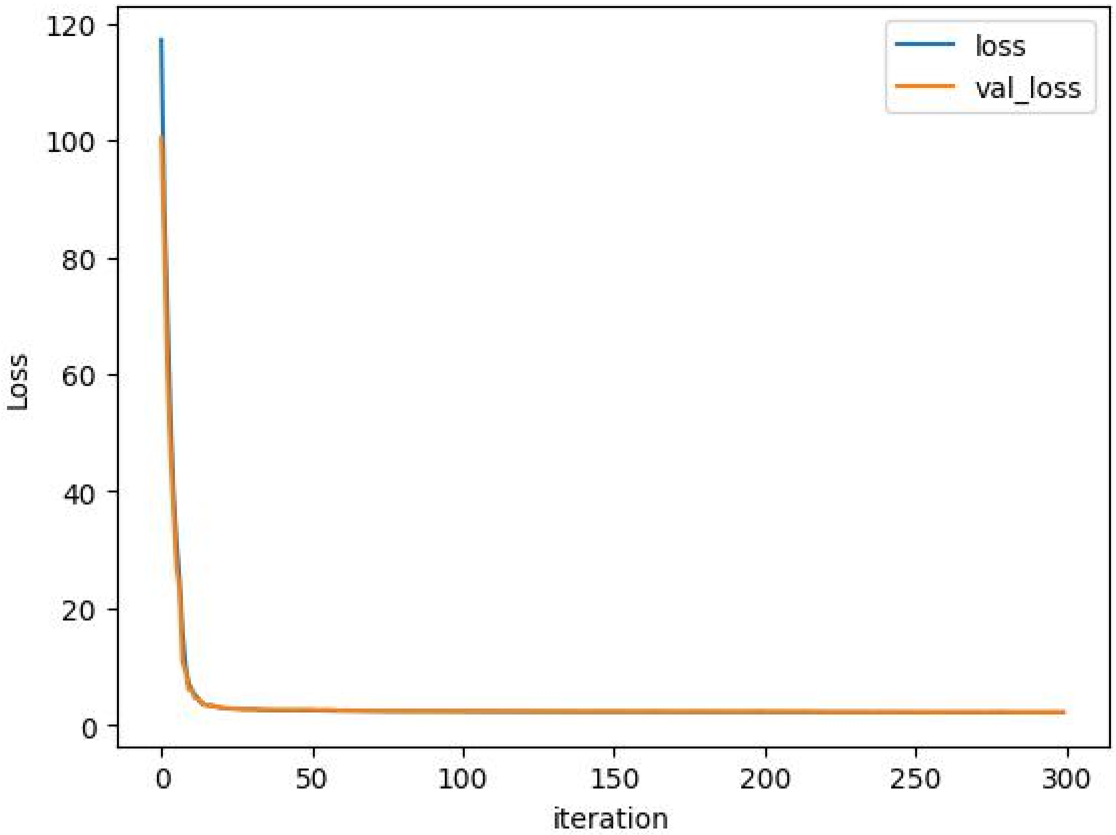




Fashion MNIST Vanishing Problem

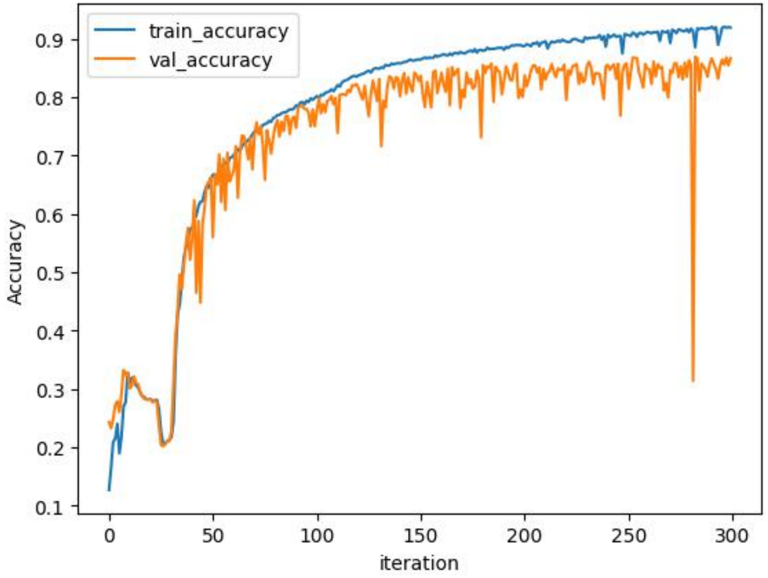
- **Weight Increasing**
  - Model:
    - **Weight Initialization**: μ=0, σ=10.0
    - **Hidden Layers**: 7 layers
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: CE
    - **Optimizer**: sgd

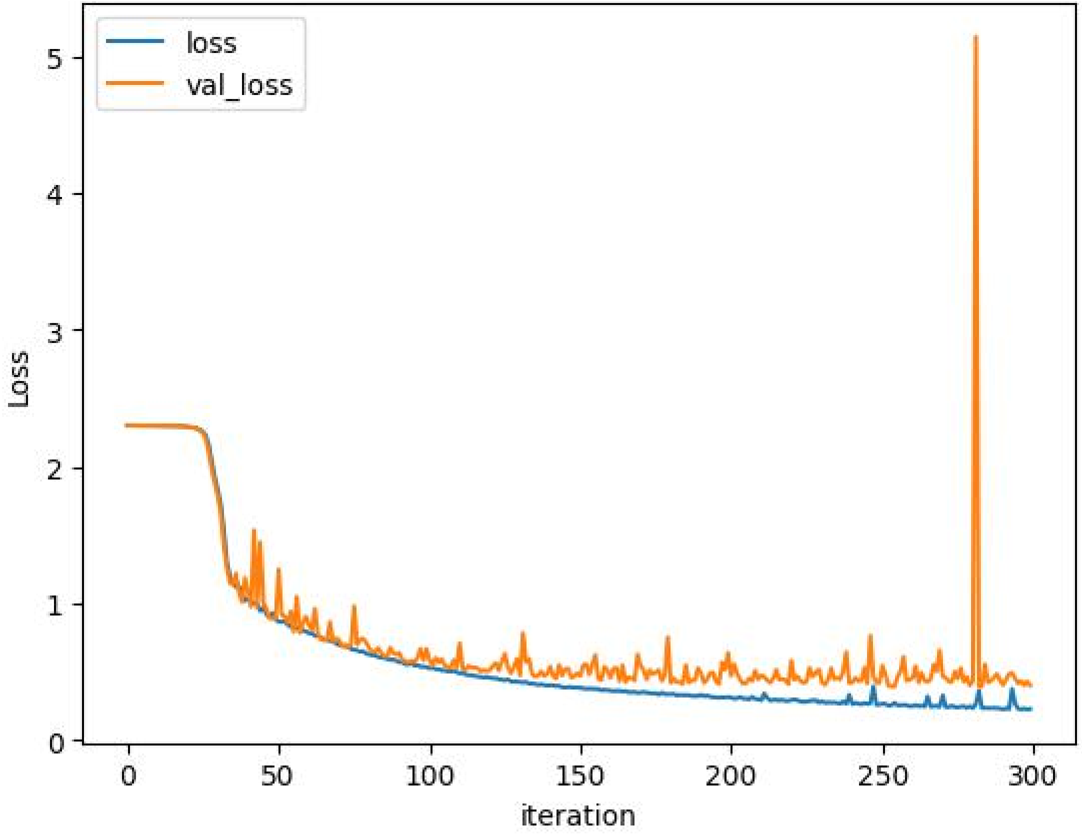




Fashion MNIST Vanishing Problem

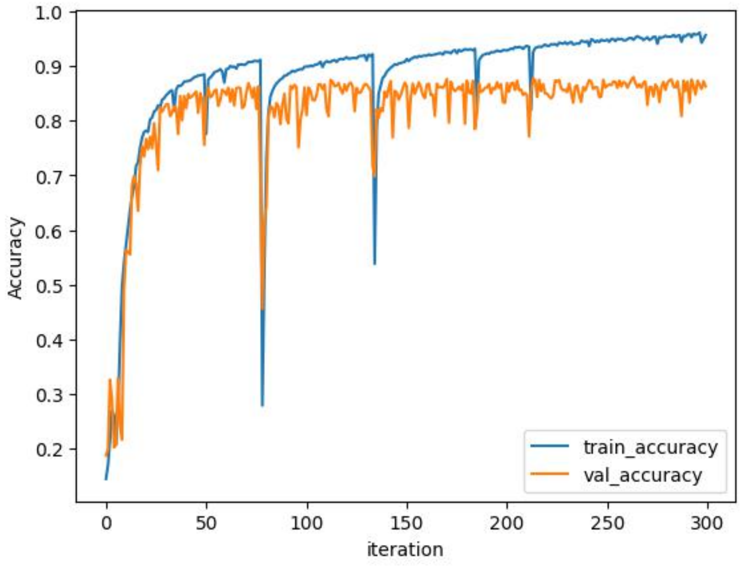
- **Better Activation**
  - Model:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: 7 layers
    - **Activation**: relu
    - **Nodes**: 128
    - **Loss**: CE
    - **Optimizer**: sgd

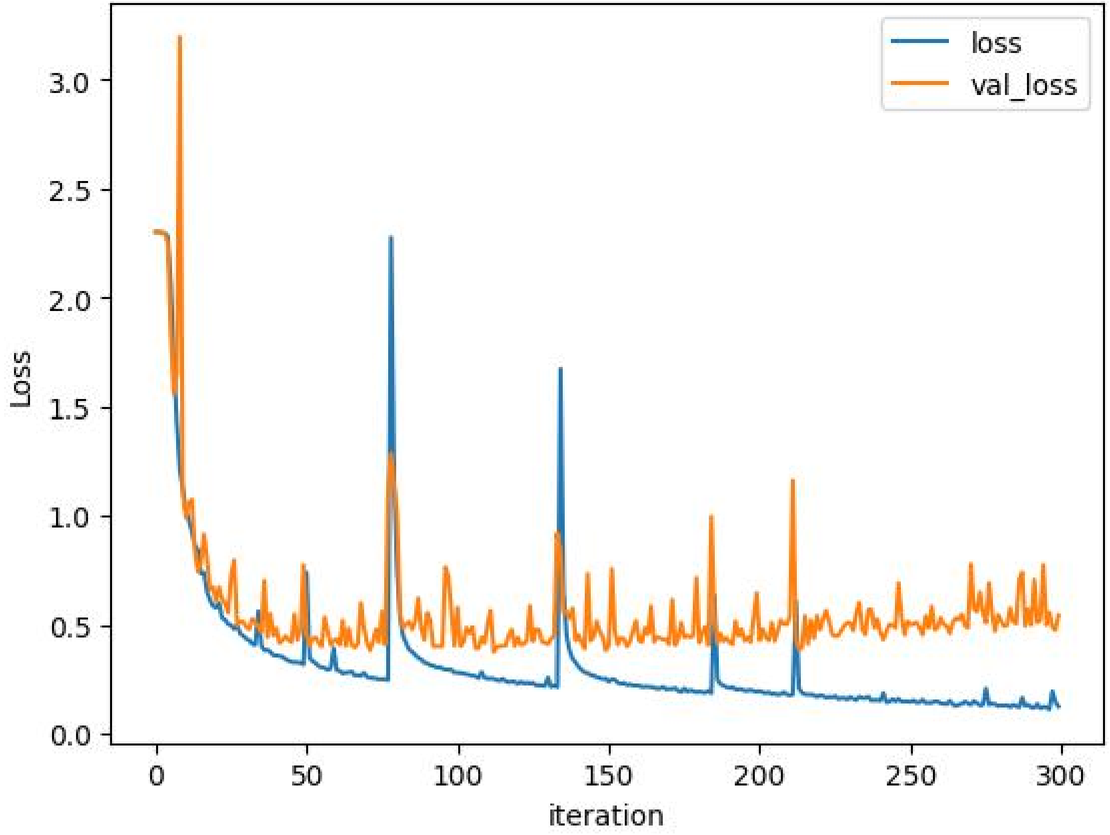




Fashion MNIST Vanishing Problem

- **Better Activation**
  - Model:
    - **Weight Initialization**: $\mu=0$, $\sigma=0.05$
    - **Hidden Layers**: 7 layers
    - **Activation**: relu
    - **Nodes**: 128
    - **Loss**: CE
    - **Optimizer**: sgd, lr=0.05

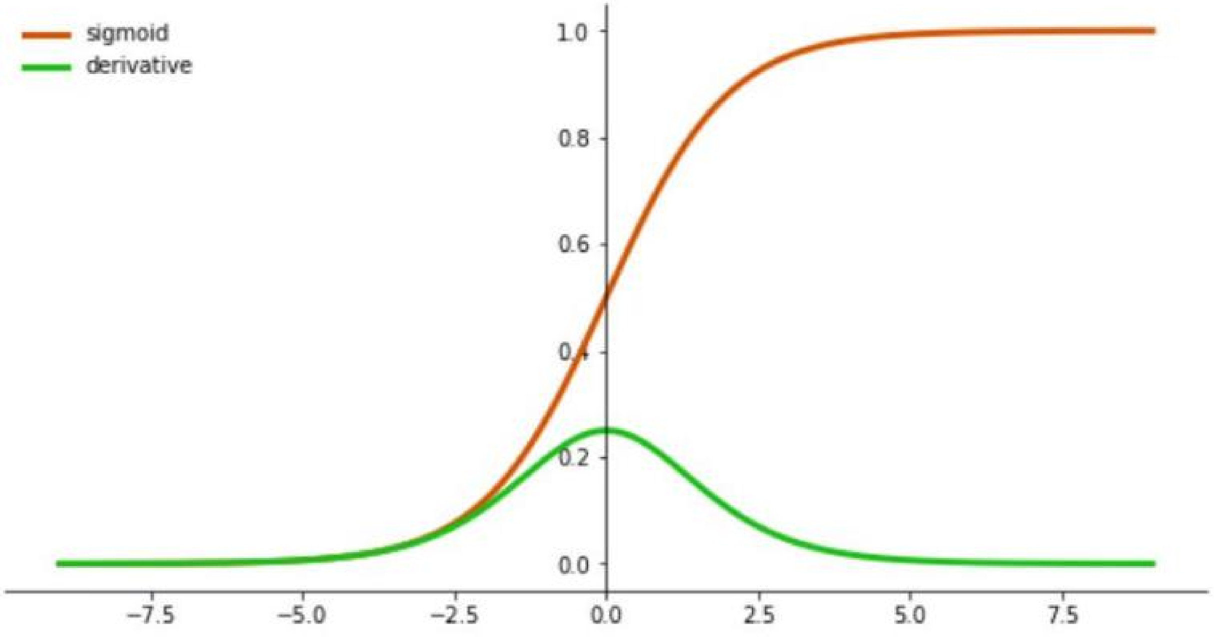




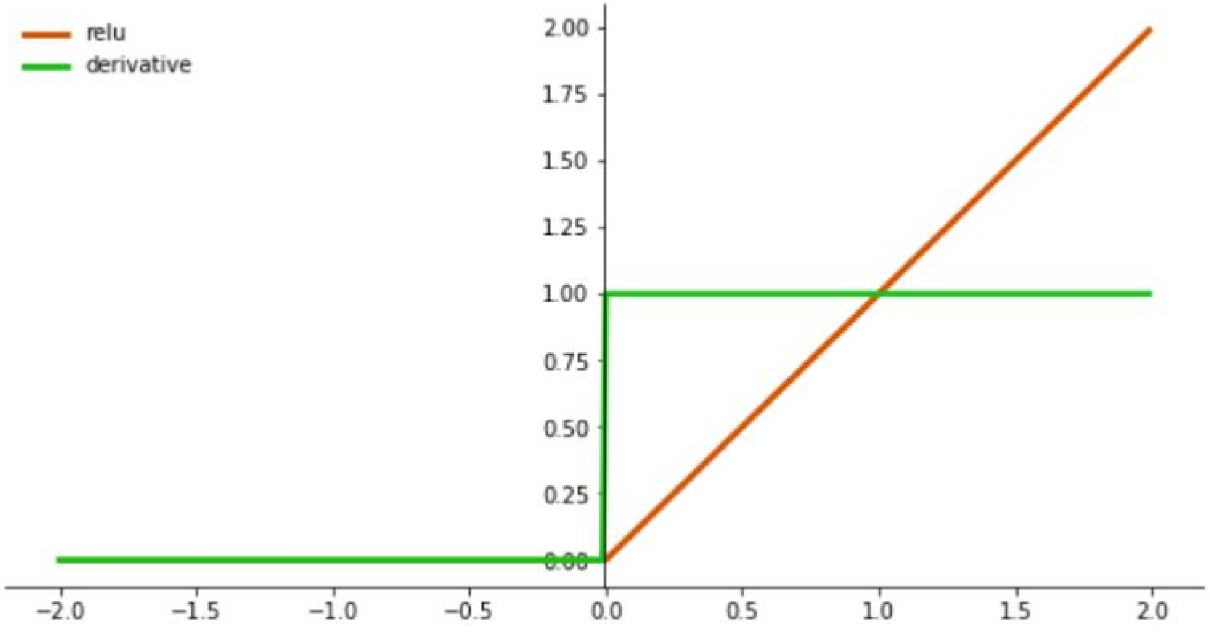
Fashion MNIST Vanishing Problem

- **Better Activation**

Sigmoid

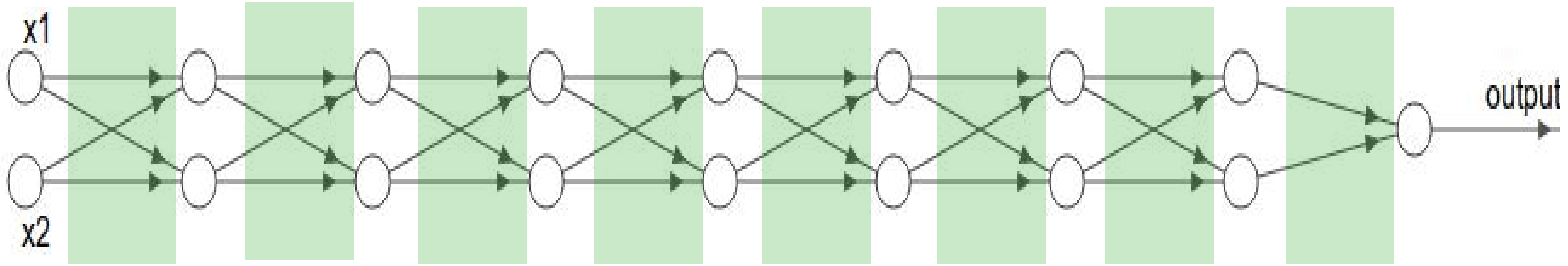


Relu



Giới thiệu Vanishing và Exploding Problem

- **Better Activation**
  – Các nguyên nhân có thể gây ra vanishing problem



$\hat{y}$

$W^{[1]}$ $W^{[2]}$ $W^{[3]}$ $W^{[L]}$

- $\frac{\partial L}{\partial w^{[L]}} = \frac{\partial L}{\partial a^{[L]}} * \frac{\partial a^{[L]}}{\partial z^{[L]}} * \frac{\partial z^{[L]}}{\partial w^{[L]}}$

- $Loss = L(\hat{y}, y), \ \hat{y} = a^{[L]}$
- $a^{[l]} = g(z^{[l]})$

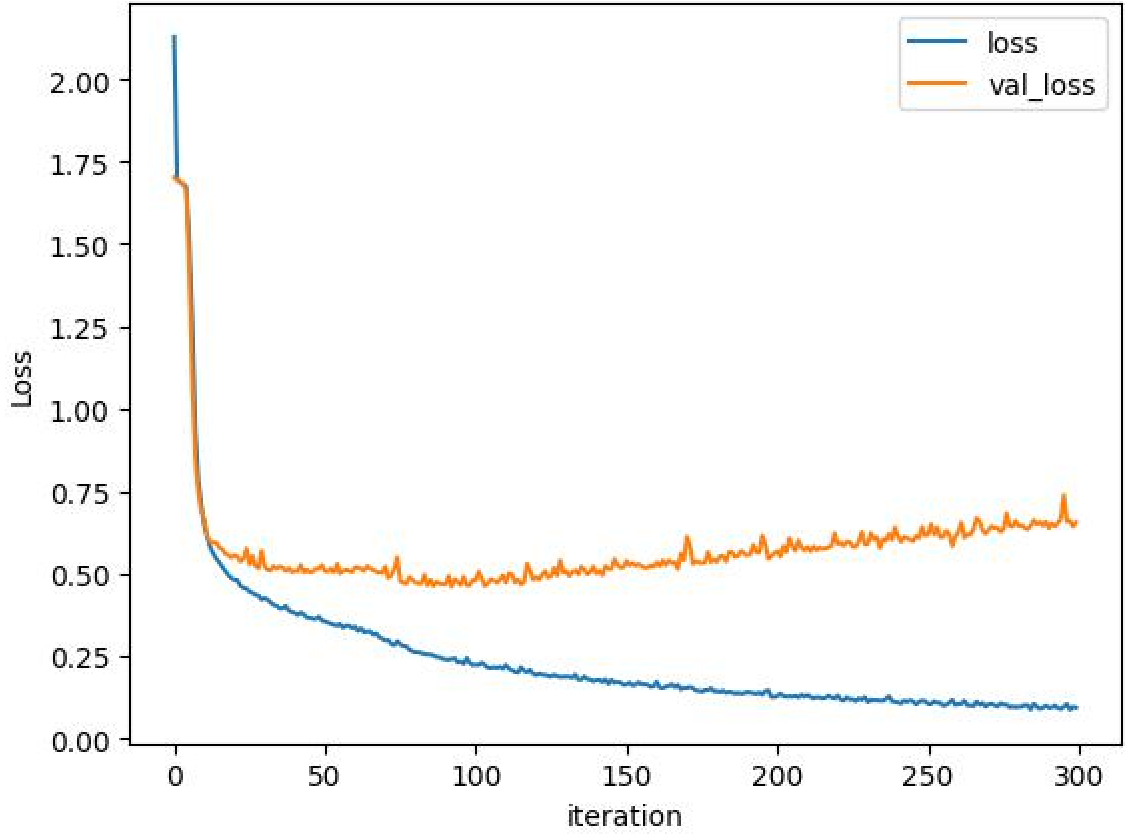
$$\boldsymbol{w^{[l]} = w^{[l]} - \eta \frac{\partial L}{\partial w^{l}}}$$

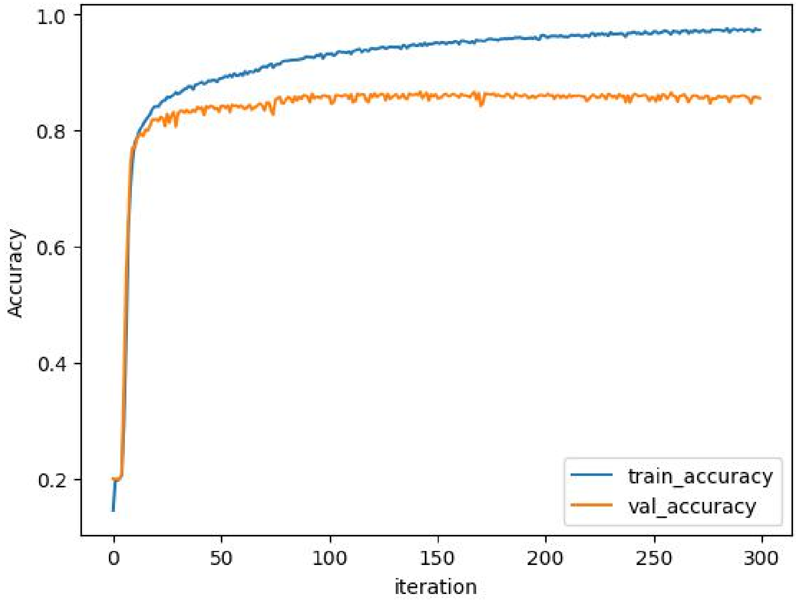
if $g(\bullet)$ là **ReLu function** $\sigma(x)$

- $\frac{\partial L}{\partial w^{[1]}} = \frac{\partial L}{\partial a^{[L]}} * \underbrace{\frac{\partial a^{[L]}}{\partial z^{[L]}}}_{\text{0 or 1}} * \frac{\partial z^{[L]}}{\partial a^{[L-1]}} \ldots * \underbrace{\frac{\partial a^{[2]}}{\partial z^{[2]}}}_{\text{0 or 1}} * \frac{\partial z^{[2]}}{\partial a^{[1]}} * \underbrace{\frac{\partial a^{[1]}}{\partial z^{[1]}}}_{\text{0 or 1}} * \frac{\partial z^{[1]}}{\partial w^{[1]}}$

Fashion MNIST Vanishing Problem

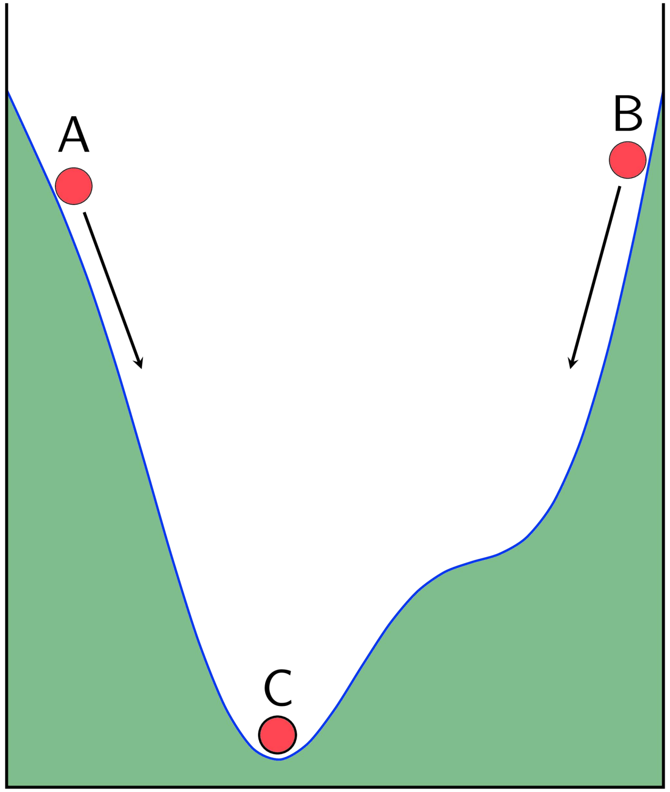
- **Better Optimizer**
  - Model:
    - **Weight Initialization**: $\mu=0$, $\sigma=0.05$
    - **Hidden Layers**: 7 layers
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: BCE
    - **Optimizer**: Adam



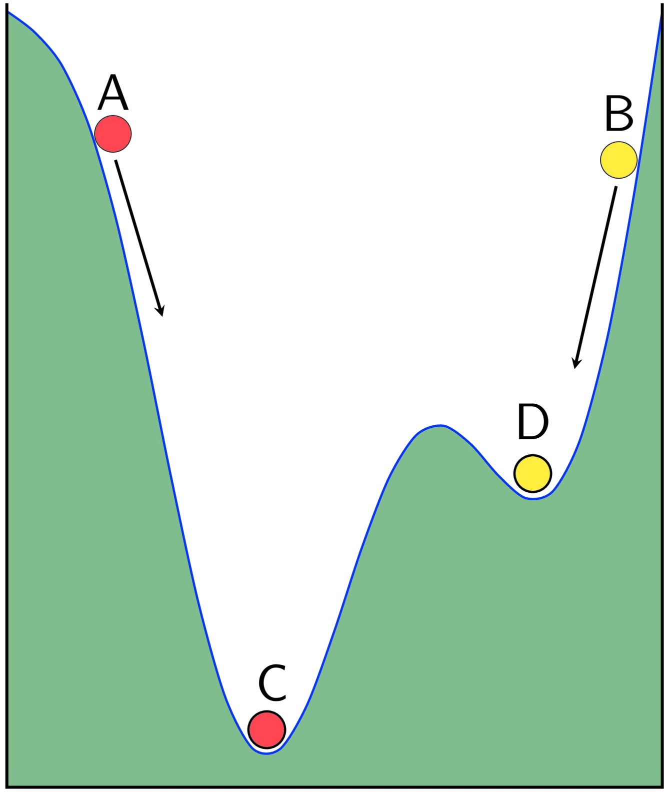


Fashion MNIST Vanishing Problem

- **Better Optimizer**
  - SGD với momentum



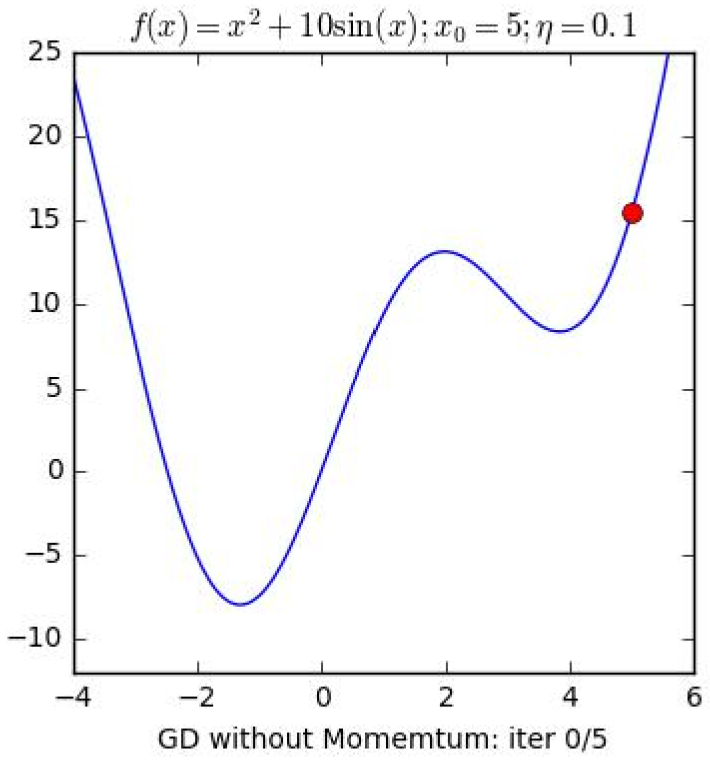
a) GD

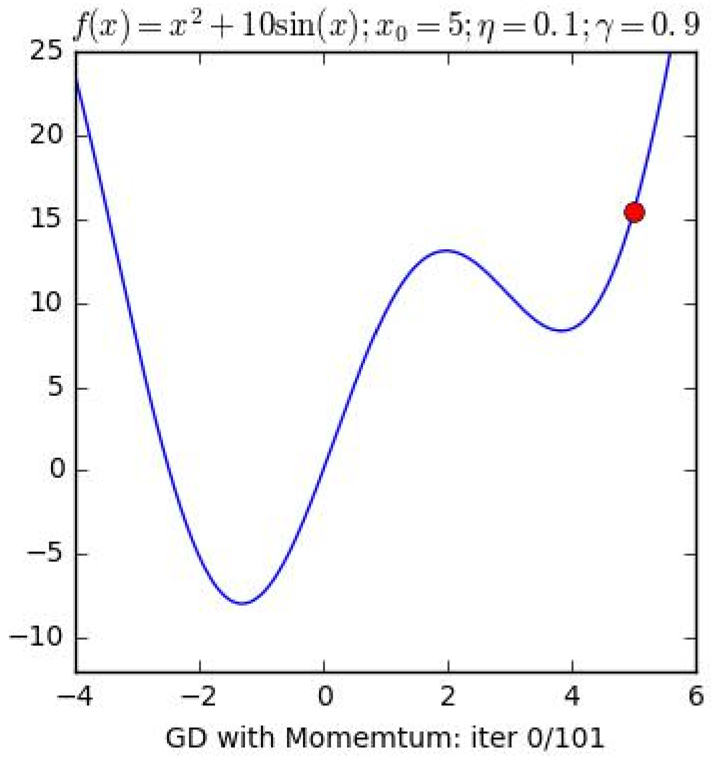


b) GD

Fashion MNIST Vanishing Problem

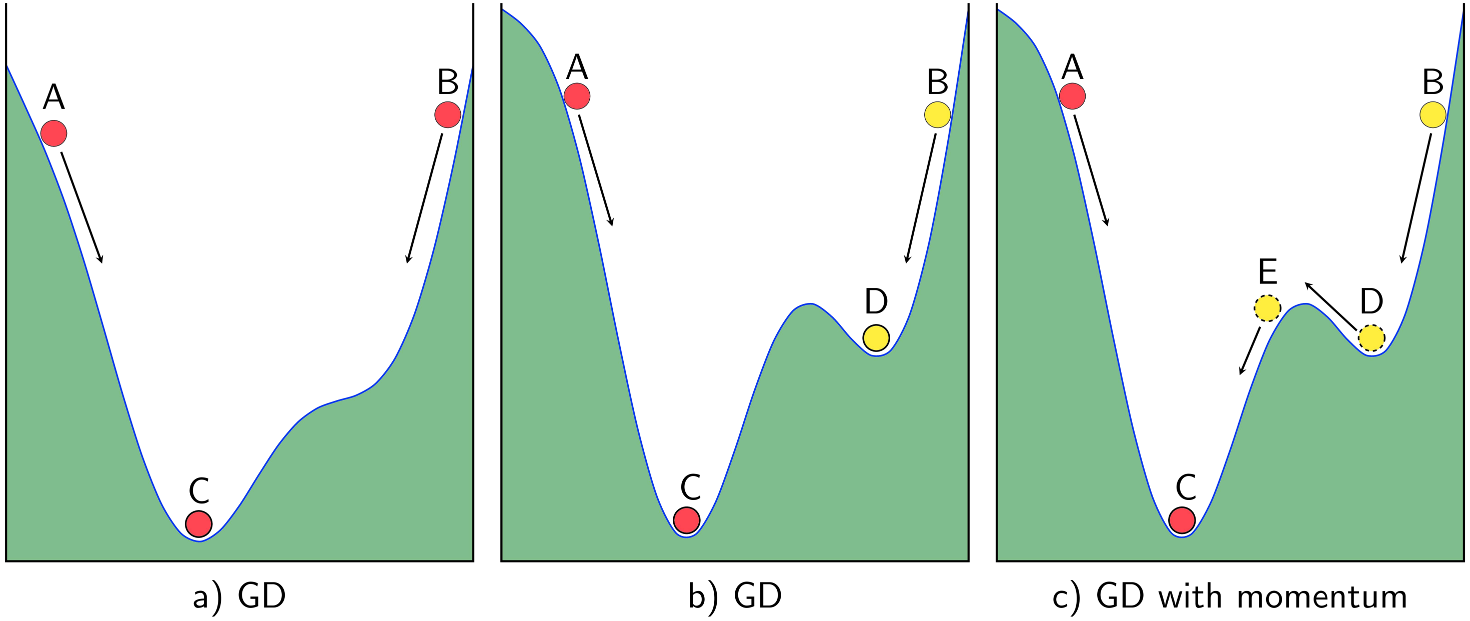
- **Better Optimizer**
  - SGD với momentum

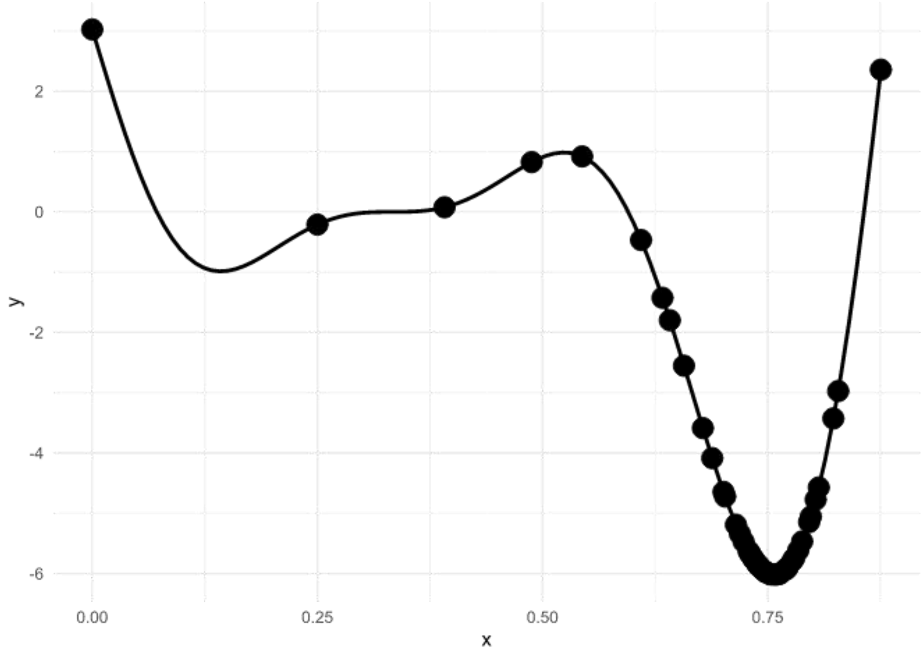




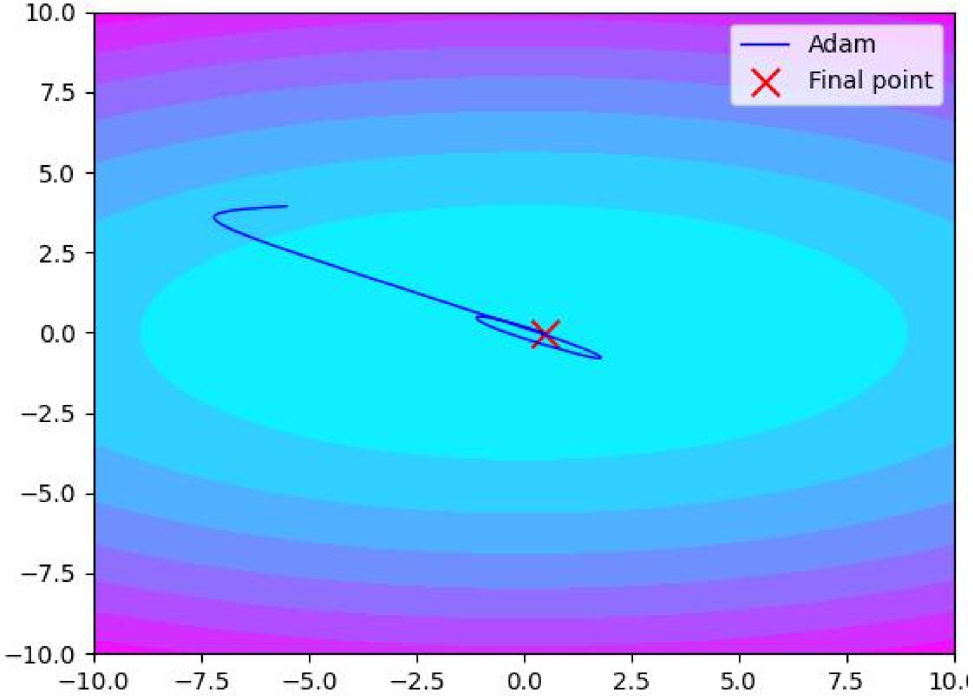
Fashion MNIST Vanishing Problem

- **Better Optimizer**
  - Adam: momentum + ma sát





- **Better Optimizer**
  - Adam: momentum + ma sát



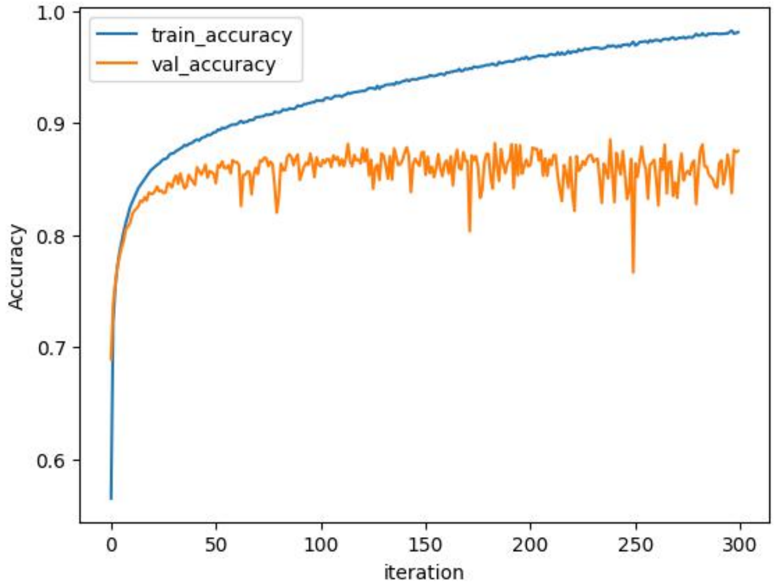
Algorithm 1: *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. g_t^2 indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t.

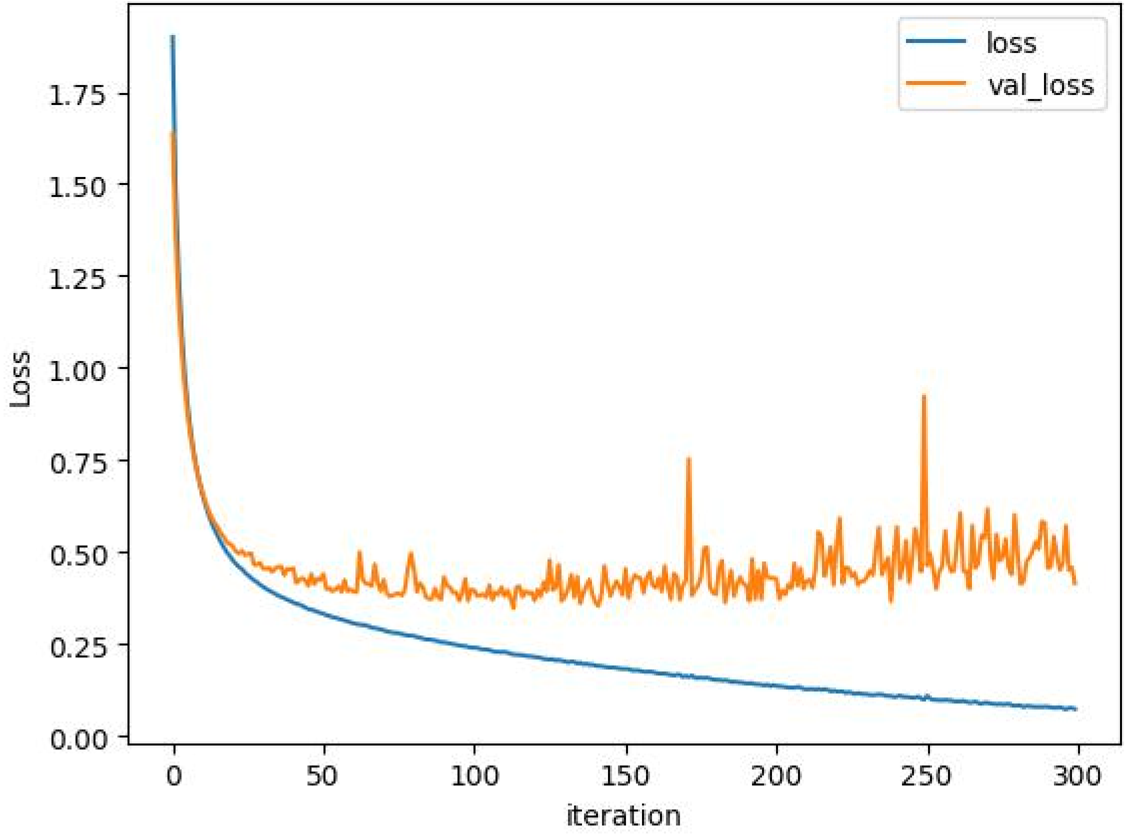
Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1st moment vector)
  $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  while θ_t not converged **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  end while
  return θ_t (Resulting parameters)

Fashion MNIST Vanishing Problem

Fashion MNIST Vanishing Problem

- **Normalize Inside Network**
  - Model:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: 7 layers + BatchNorm
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: BCE
    - **Optimizer**: sgd





Fashion MNIST Vanishing Problem

- **Normalize Inside Network**
  - BatchNormalization:
    - Giúp việc **học nhanh** hơn và **ổn định** hơn
    - Train và Test phase hoạt động khác nhau

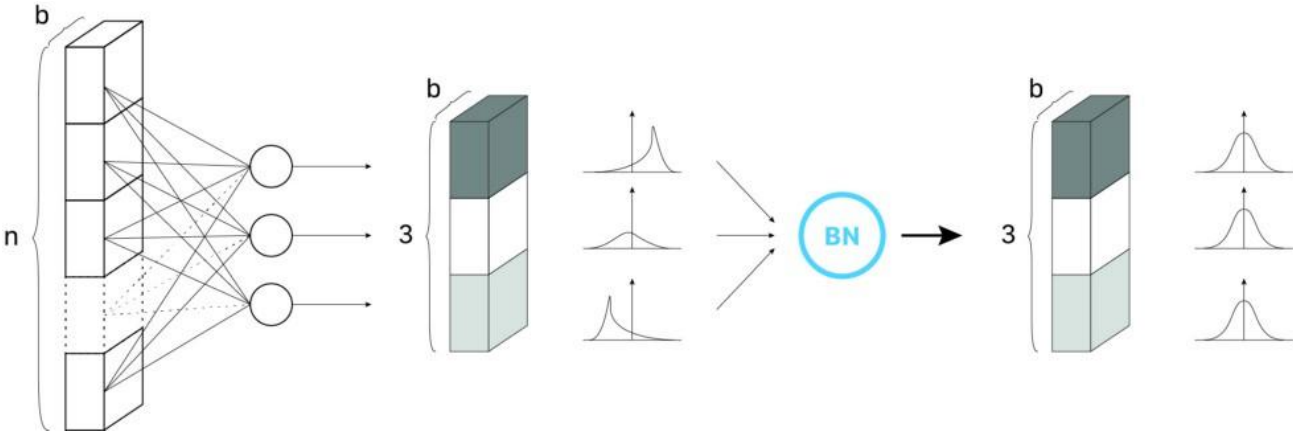
Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

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

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

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

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



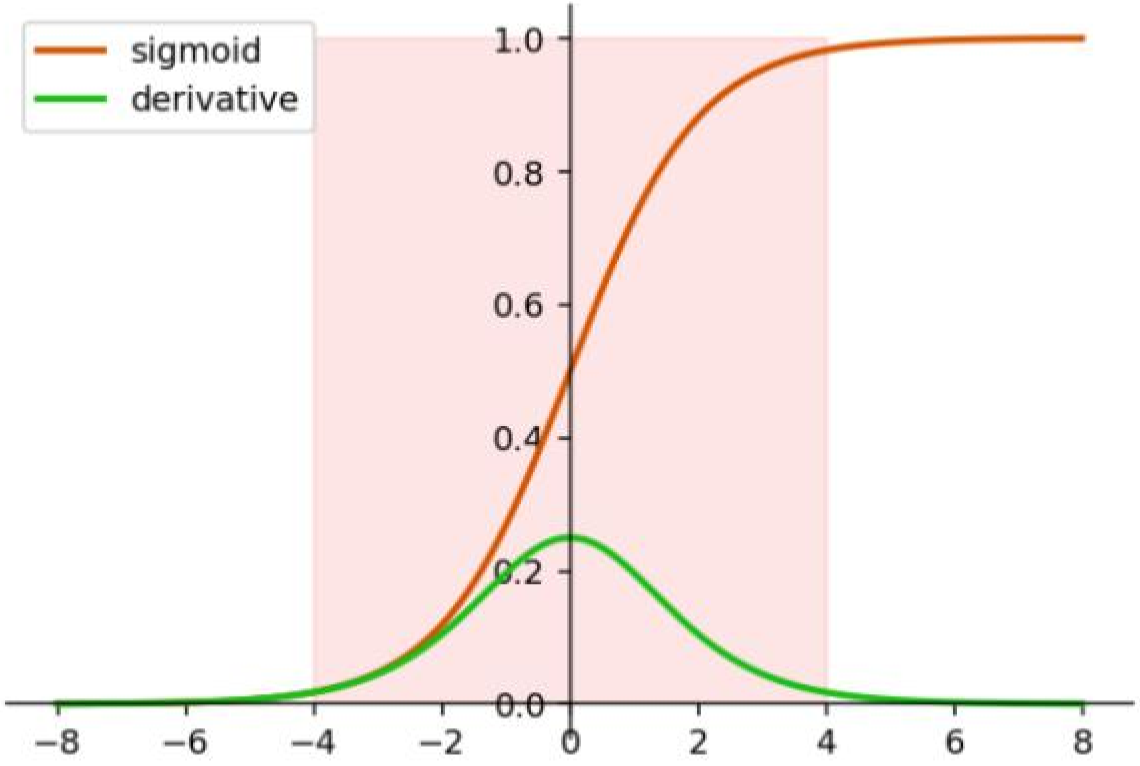
(1)(2) Tính mean và variance của 1 batch

(3) normalize để mỗi node output theo 1 normal distribtuion

(4) γ và β là 2 tham số học trong train phase để scale và ship distribtuion

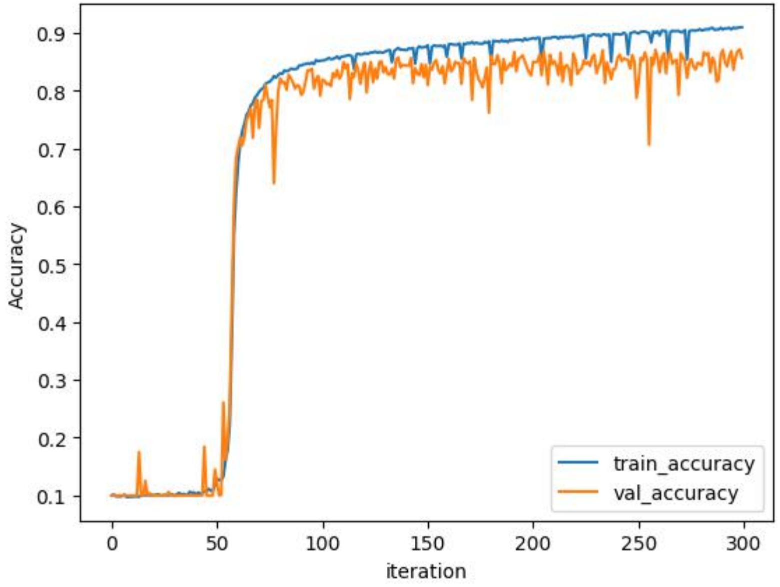
Fashion MNIST Vanishing Problem

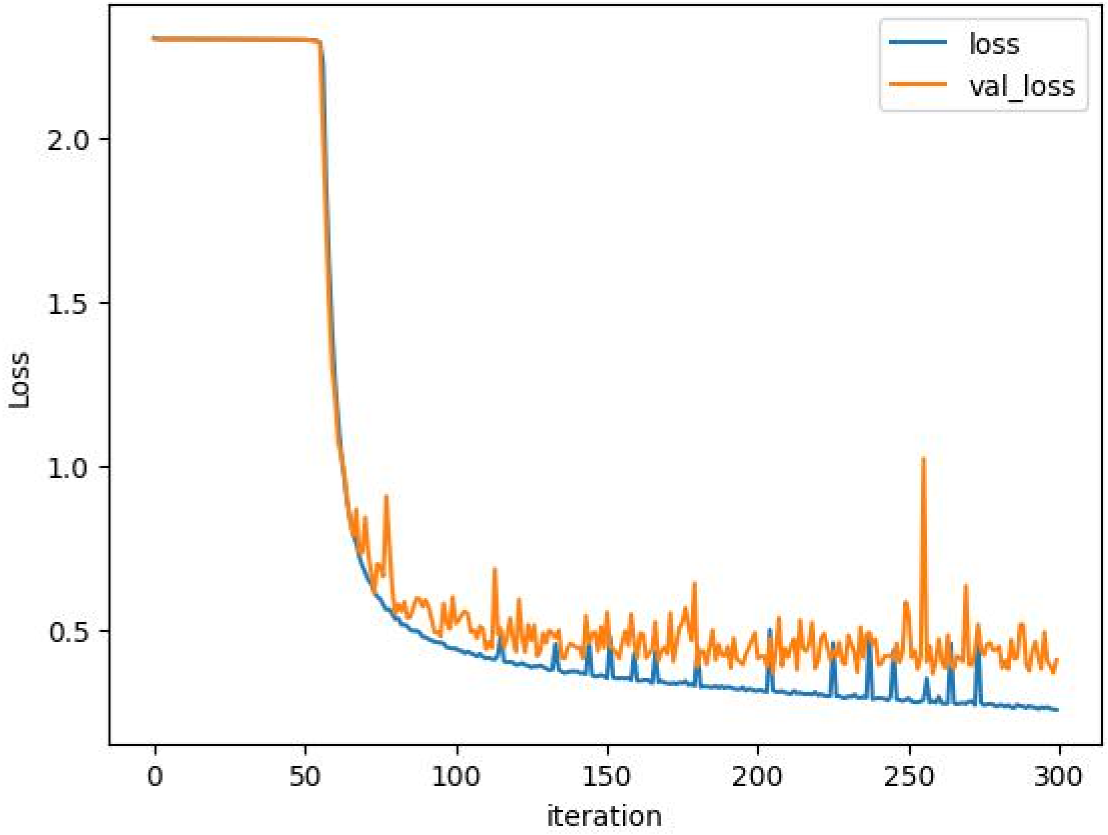
- **Normalize Inside Network**
  - Problem: input có giá trị càng lớn sẽ càng bị giới hạn và tại vị trí đó dường như không có đạo hàm
  - Sử dụng BatchNormalization giữ hoạt động trong range màu xanh [-4,4] nơi có đạo hàm mạnh



Fashion MNIST Vanishing Problem

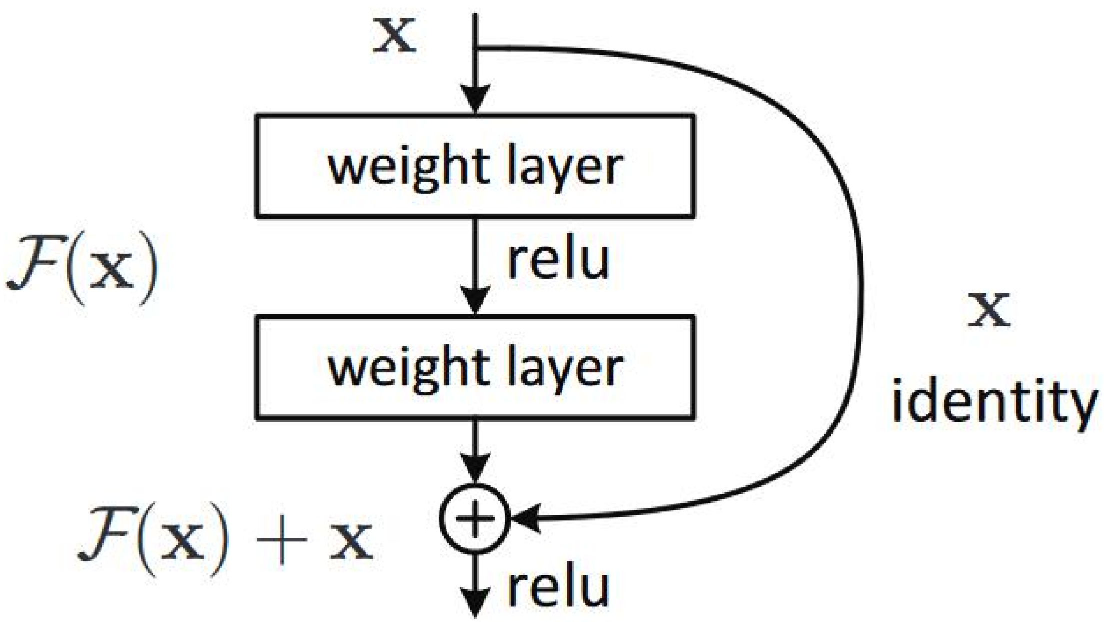
- **Normalize Inside Network**
  - Model:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: 7 layers + CustomNorm
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: BCE
    - **Optimizer**: sgd





Fashion MNIST Vanishing Problem

- **Skip Connection**



$$H(x) = F(x) + x$$

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial H}\frac{\partial H}{\partial x} = \frac{\partial L}{\partial H}\left(\frac{\partial F}{\partial x} + 1\right) = \frac{\partial L}{\partial H}\frac{\partial F}{\partial x} + \frac{\partial L}{\partial H}$$

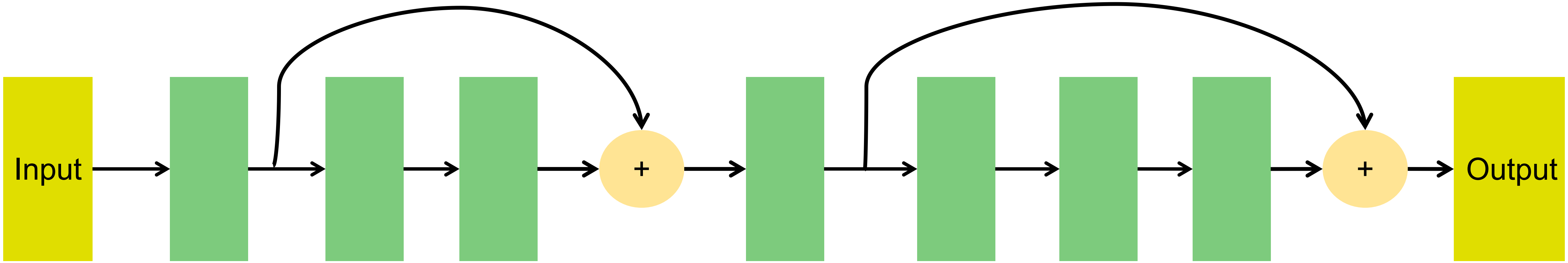
- Khi không cần học ở nhóm layer này thì nó sẽ được điều hướng và học như identity function

- Gradient qua các layer có thể sẽ nhỏ dần và bằng 0, do đó skip connection giup thông tin truyền ngược lại dễ hơn

Deep Residual Learning for Image Recognition, Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

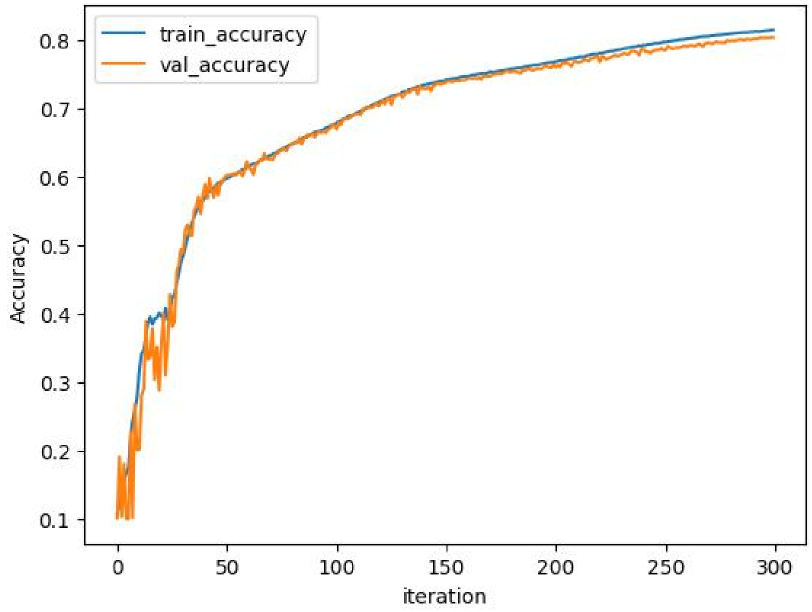
Fashion MNIST Vanishing Problem

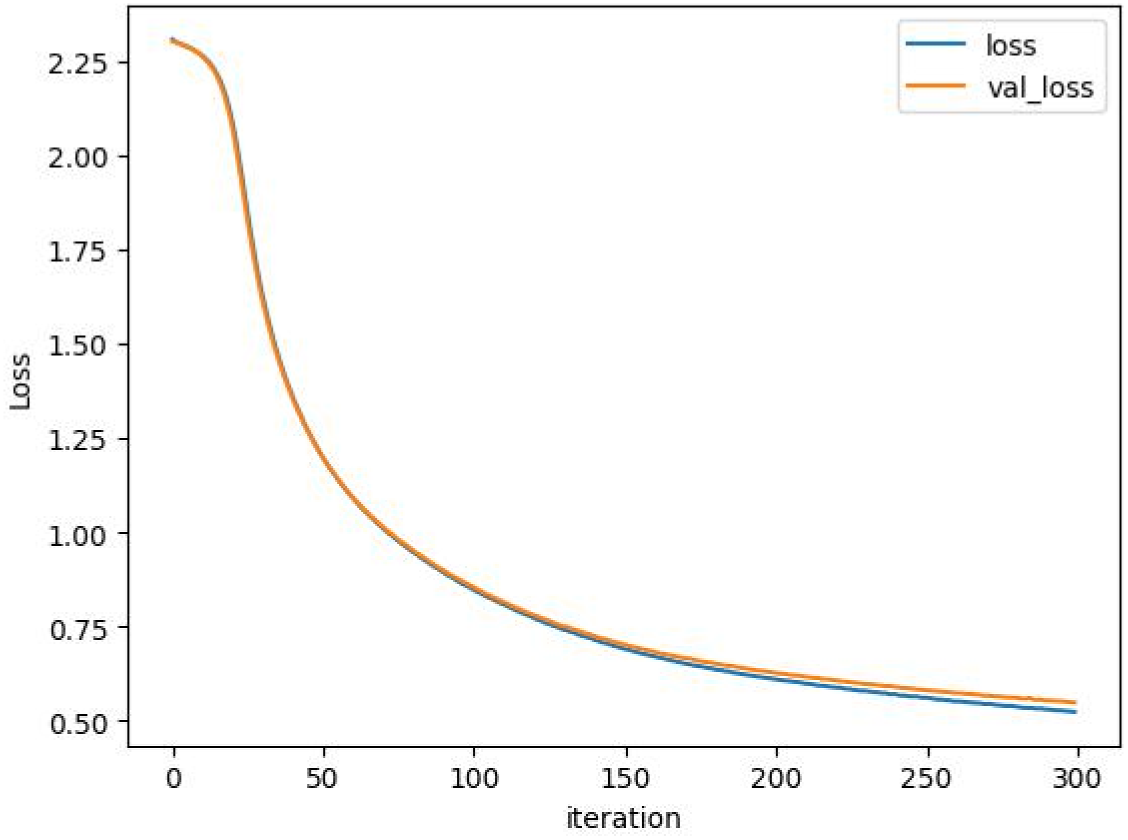
- **Skip Connection**
  - Model1:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: 7 layers + SkipConnection
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: BCE
    - **Optimizer**: sgd



Fashion MNIST Vanishing Problem

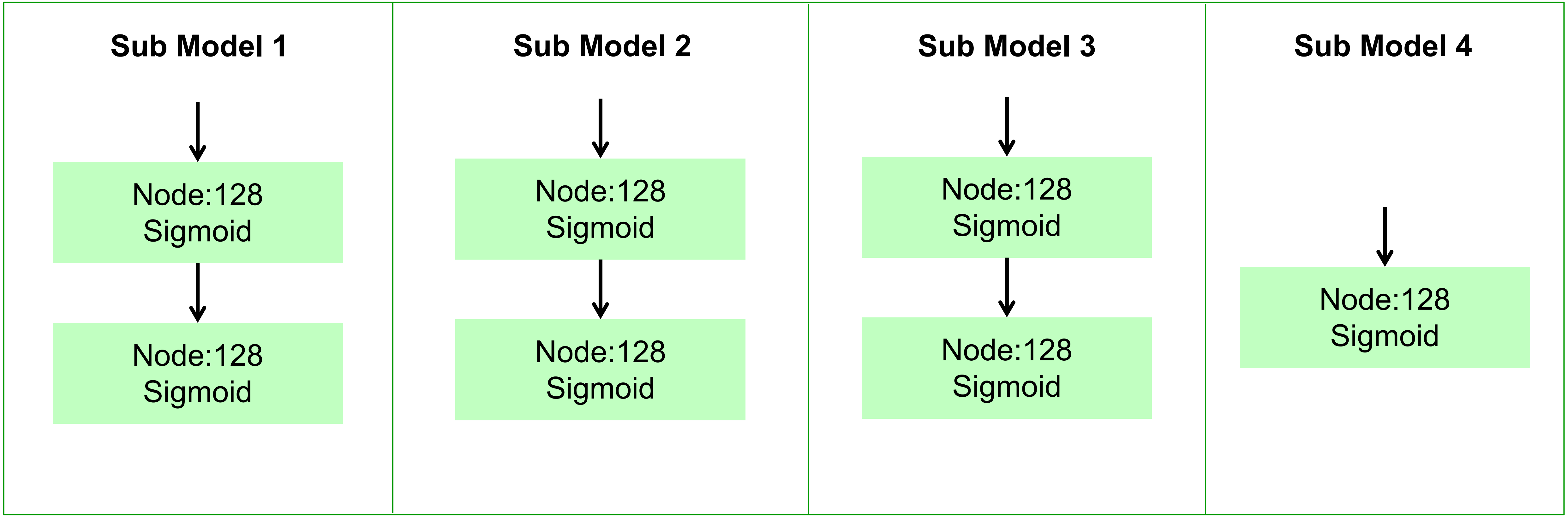
- **Skip Connection**
  - Model:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: 7 layers + SkipConnection
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: BCE
    - **Optimizer**: sgd





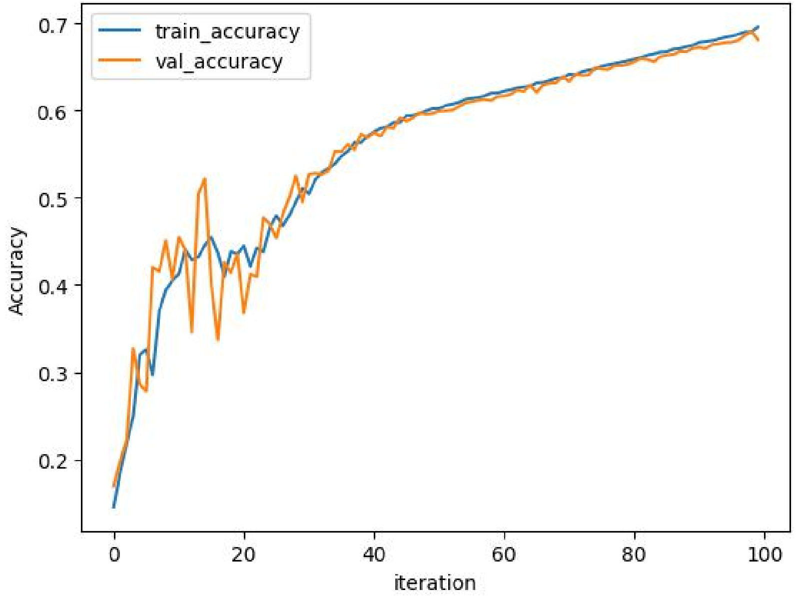
Fashion MNIST Vanishing Problem

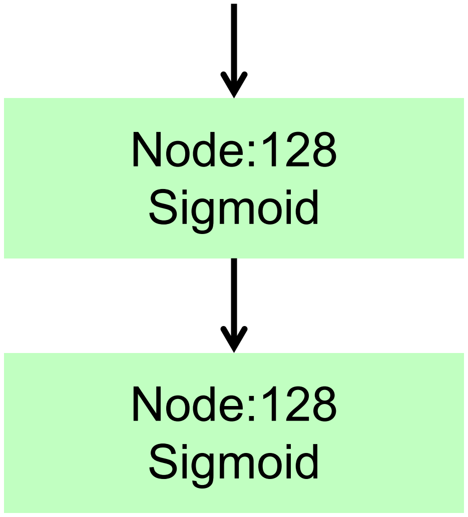
- **Train Some Layer**

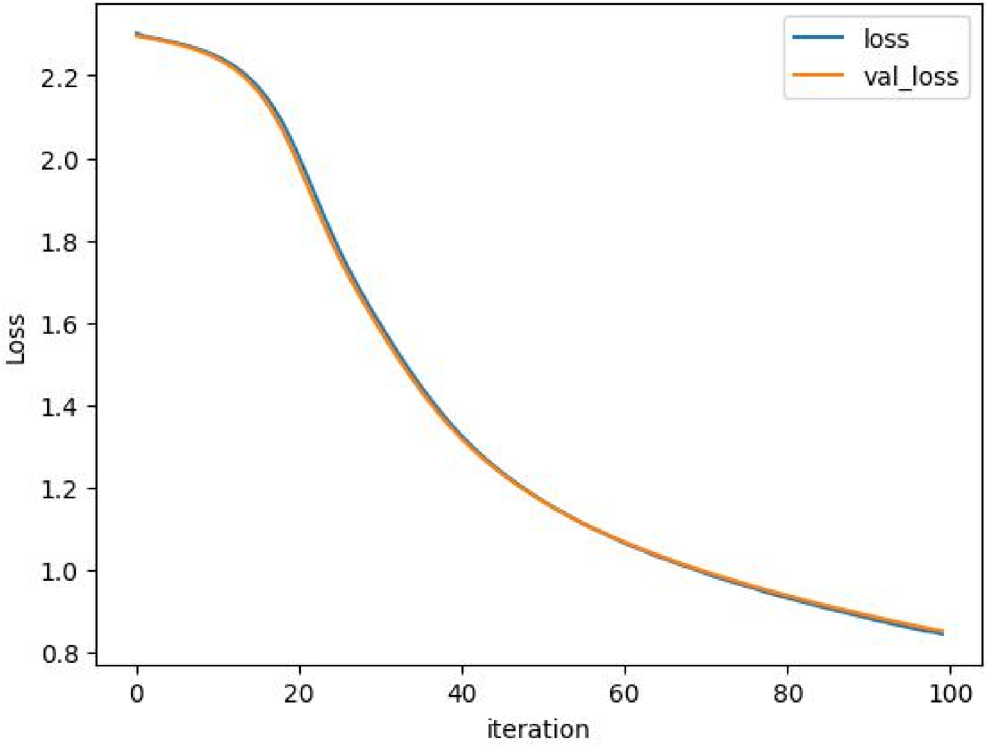


Fashion MNIST Vanishing Problem

- **Train Some Layer**
  - Train lần 1:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: sub model1 (2 layers)
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: BCE
    - **Optimizer**: sgd

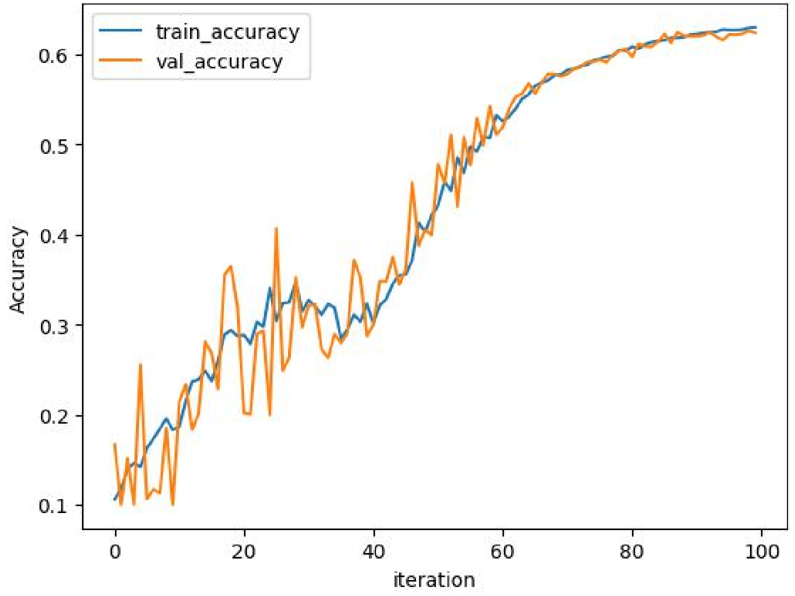


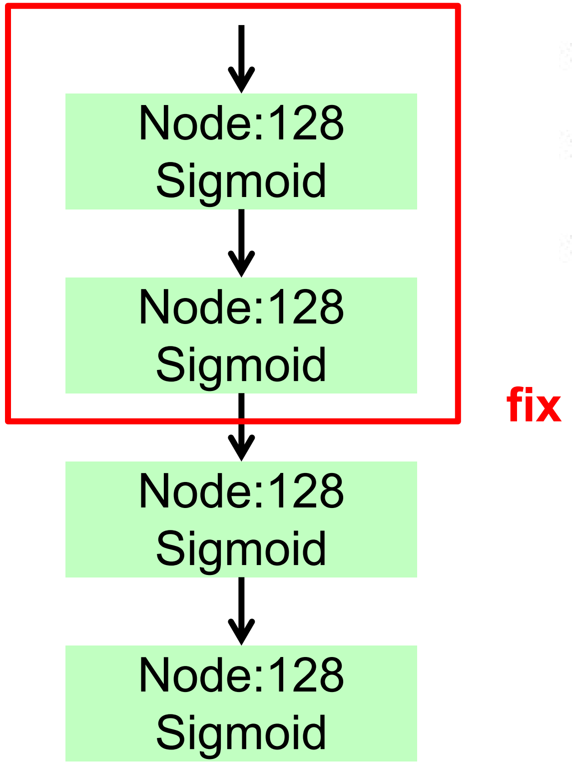


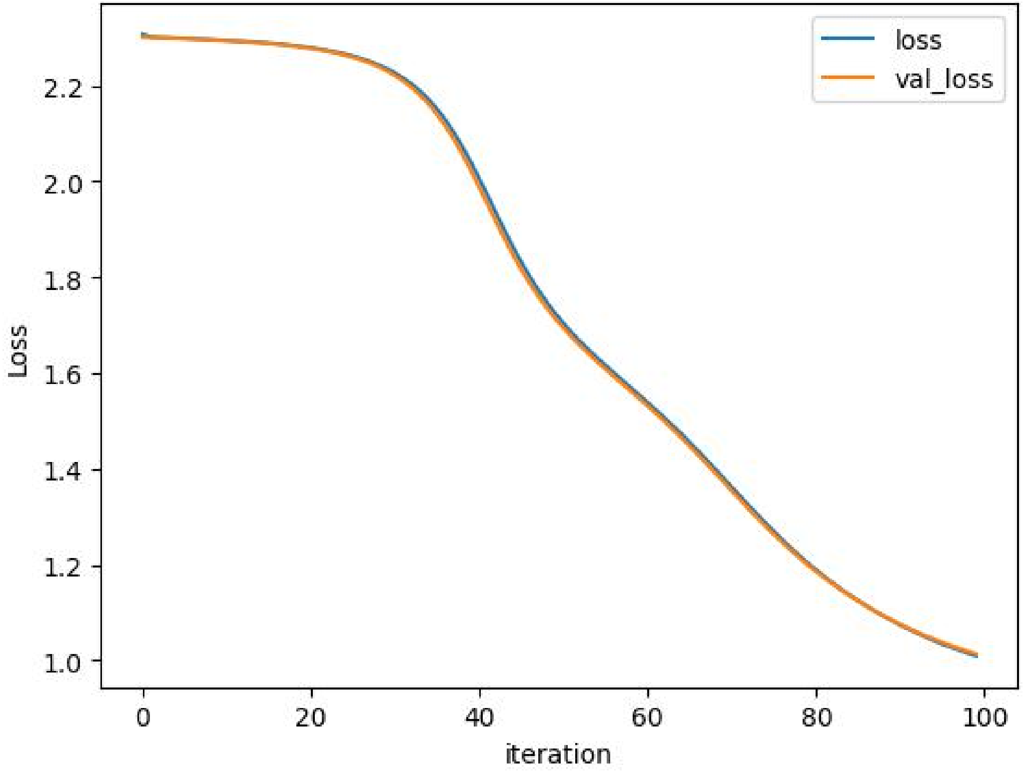


Fashion MNIST Vanishing Problem

- **Train Some Layer**
  - Train lần 2:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: sub model1(fix) + sub model2(train)
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: BCE
    - **Optimizer**: sgd

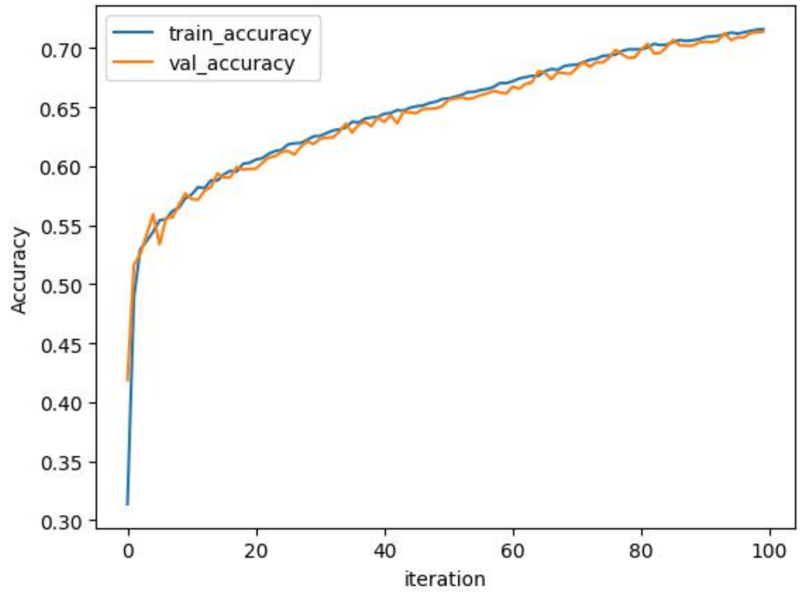


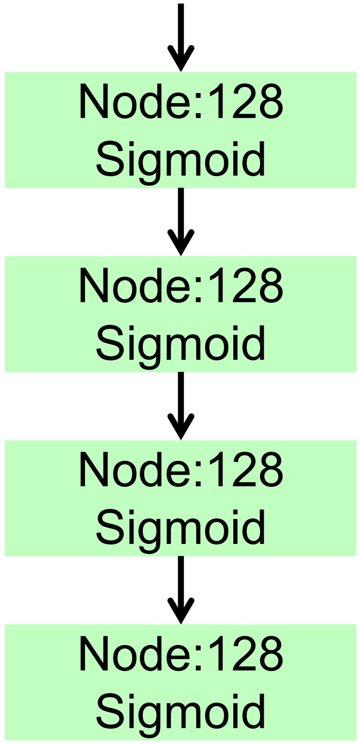


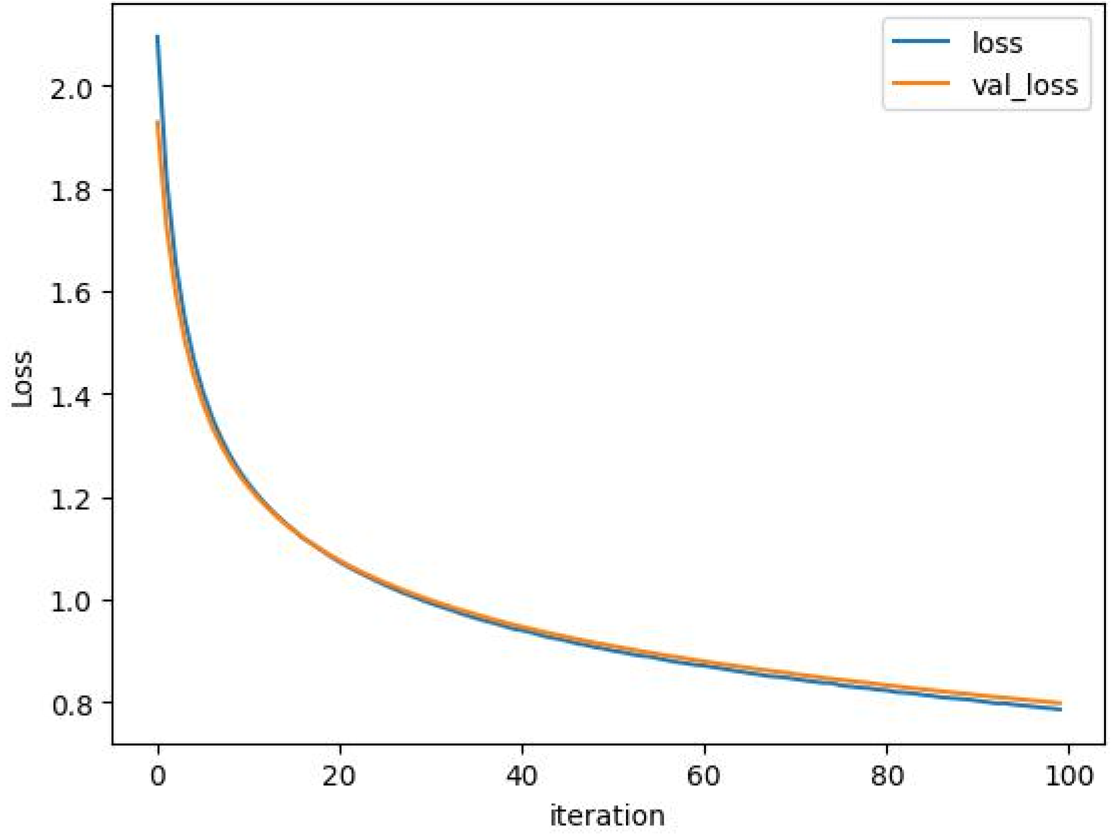


Fashion MNIST Vanishing Problem

- **Train Some Layer**
  - Train lần 3:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: sub model1(train) + sub model2(trair
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: BCE
    - **Optimizer**: sgd

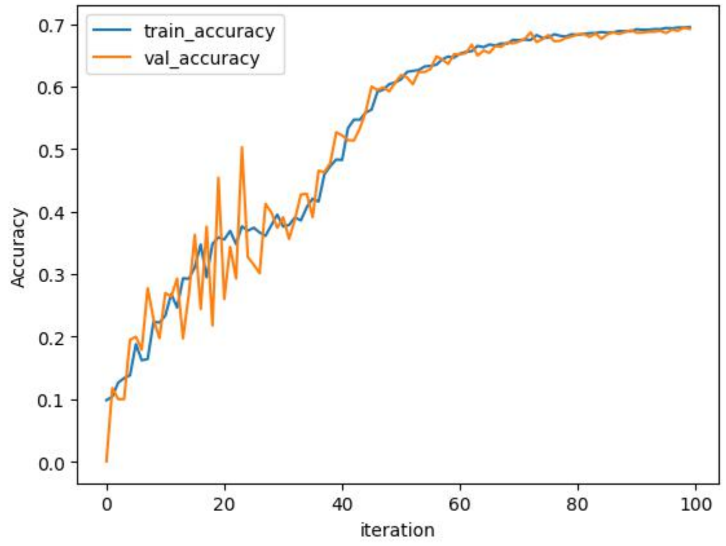


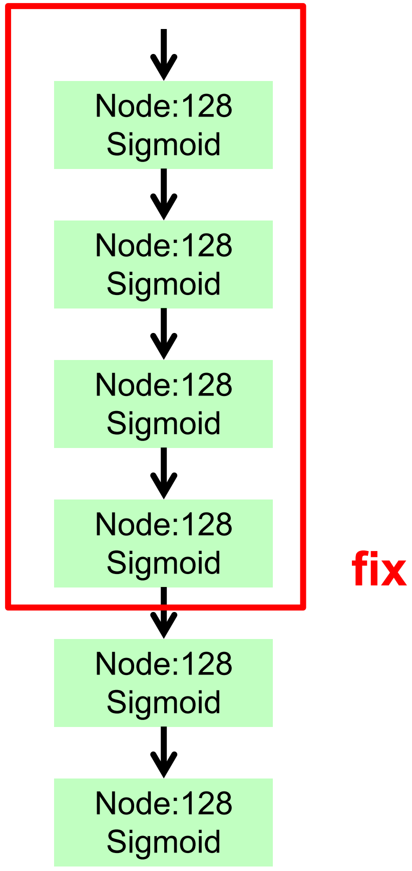


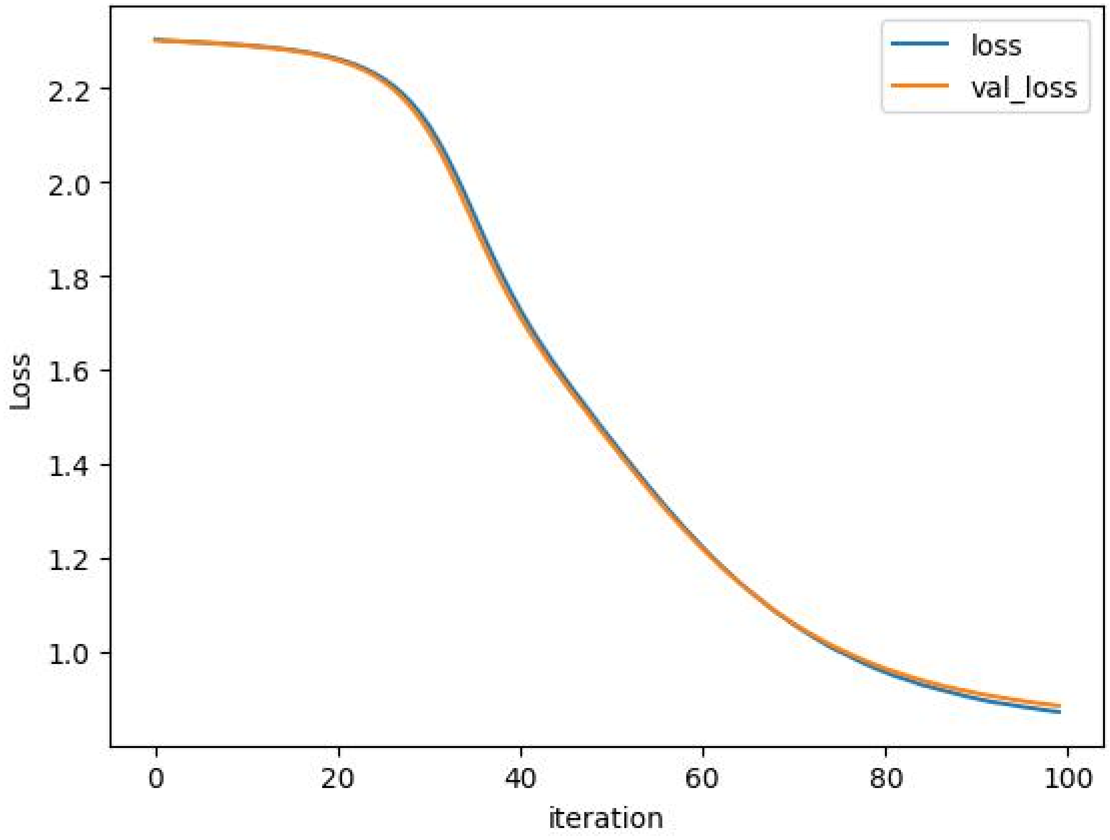


Fashion MNIST Vanishing Problem

- **Train Some Layer**
  - Train lần 4:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: sub model1(fix) + sub model2(fix) + sub model3(train)
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: BCE
    - **Optimizer**: sgd

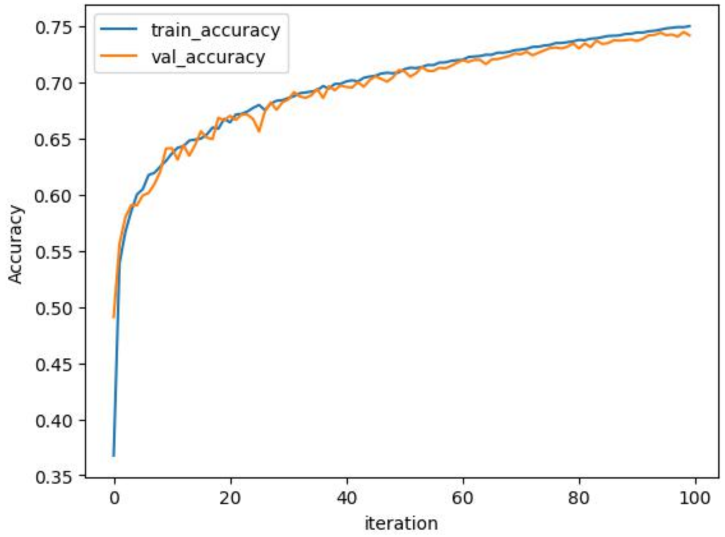


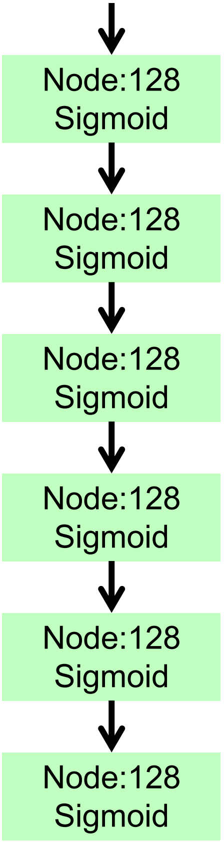


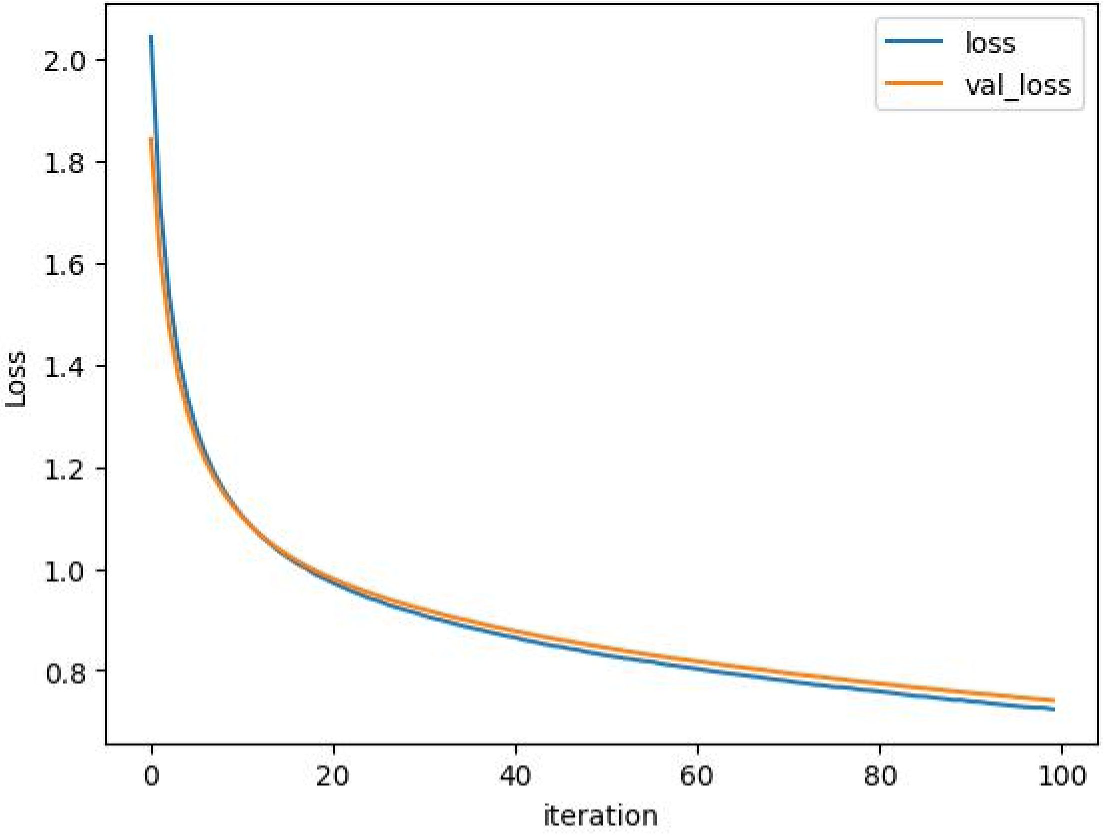


Fashion MNIST Vanishing Problem

- **Train Some Layer**
  - Train lần 5:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: sub model1(train) + sub model2(train) + sub model3(train)
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: BCE
    - **Optimizer**: sgd

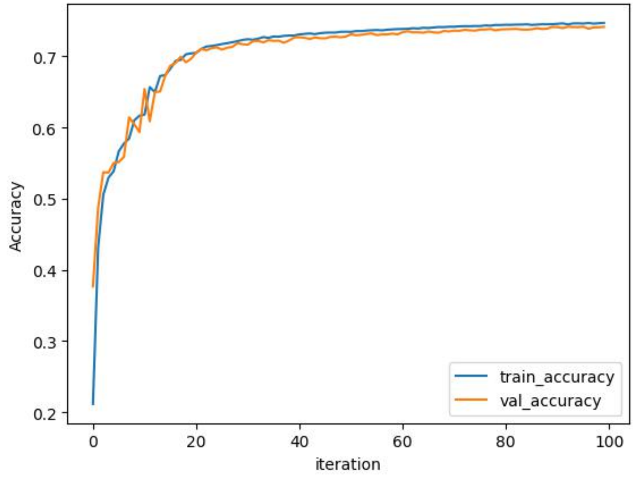


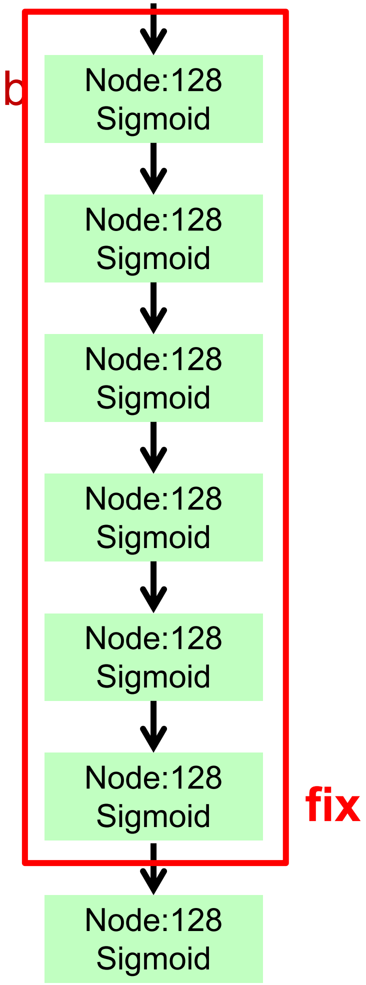


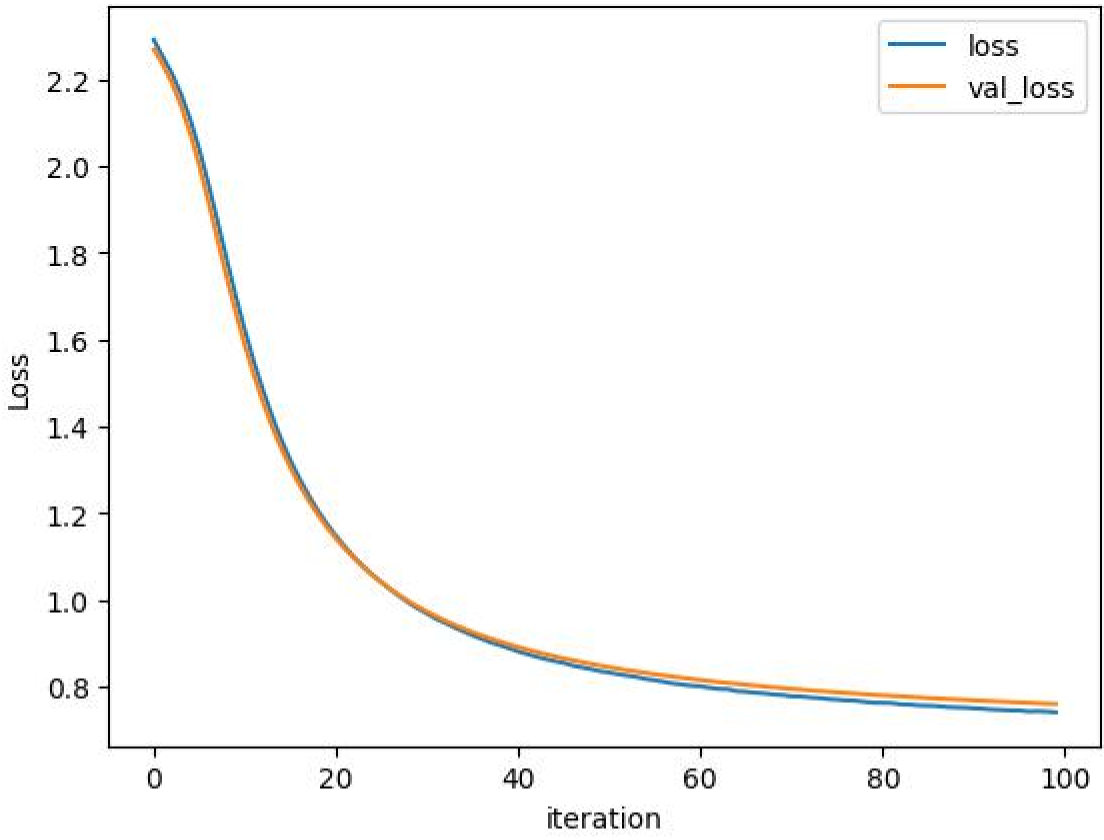


Fashion MNIST Vanishing Problem

- **Train Some Layer**
  - Train lần 6:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: sub model1(fix) + sub model2(fix) + sub model3(fix) + sub model4(train)
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: BCE
    - **Optimizer**: sgd

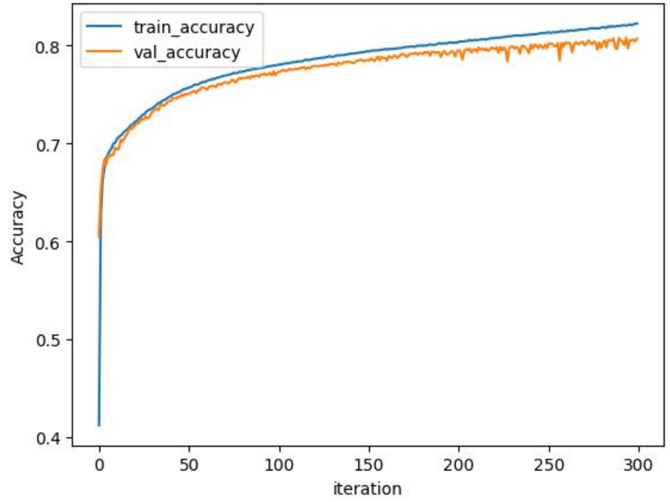


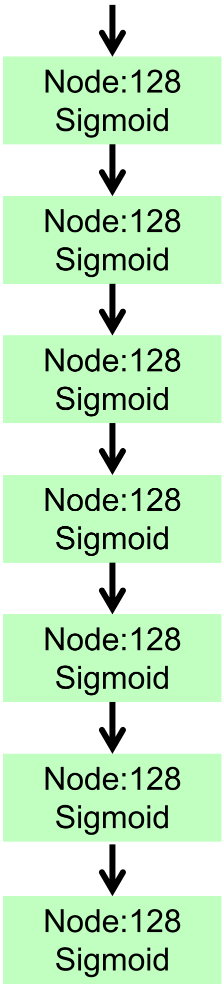


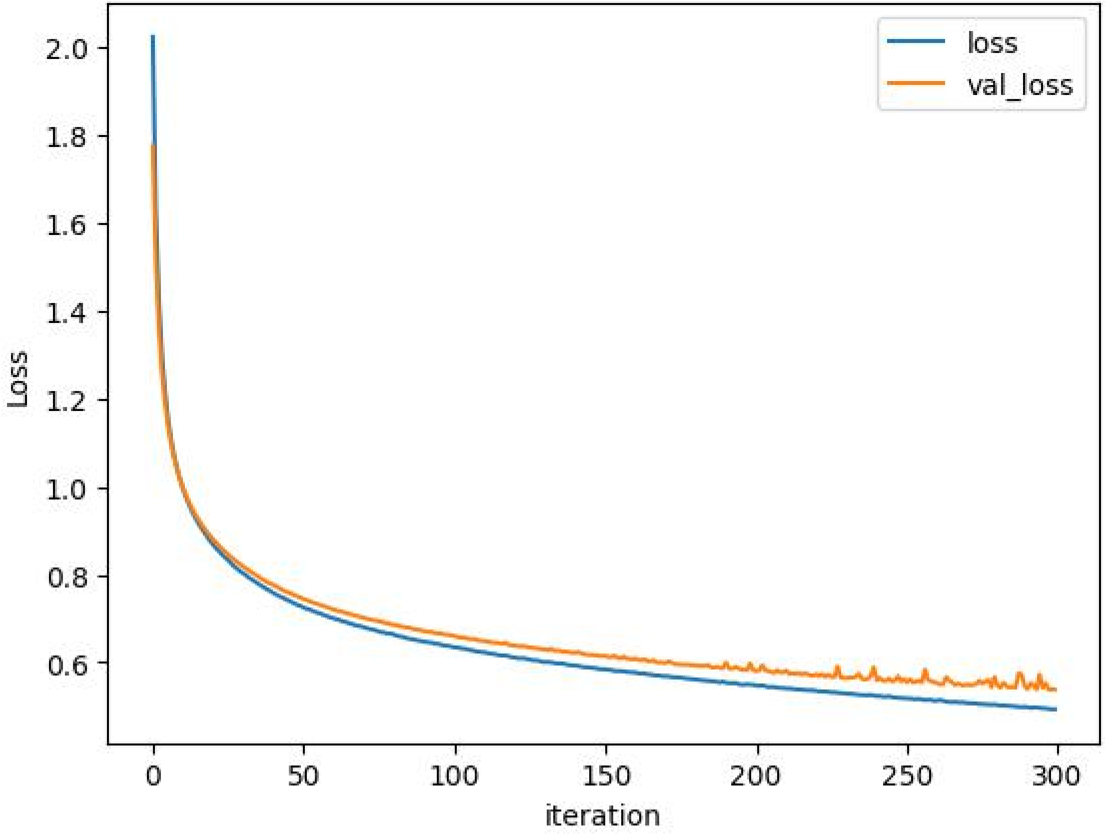


Fashion MNIST Vanishing Problem

- **Train Some Layer**
  - Train lần 7:
    - **Weight Initialization**: μ=0, σ=0.05
    - **Hidden Layers**: sub model1(train) + sub model2(train) + sub model3(train) + sub model4(train)
    - **Activation**: sigmoid
    - **Nodes**: 128
    - **Loss**: BCE
    - **Optimizer**: sgd



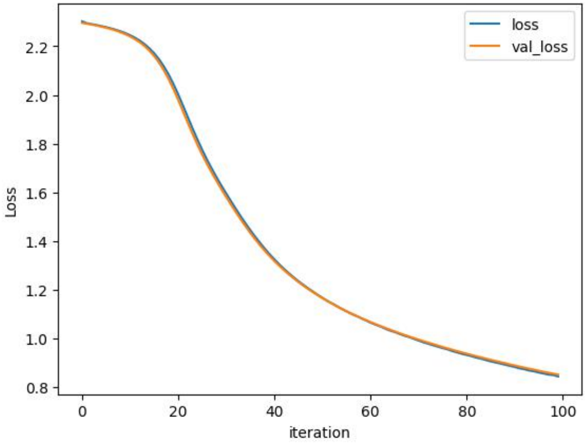




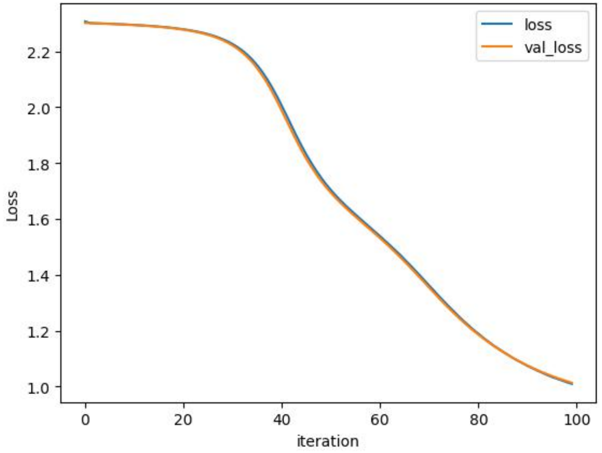
Fashion MNIST Vanishing Problem

- **Train Some Layer**

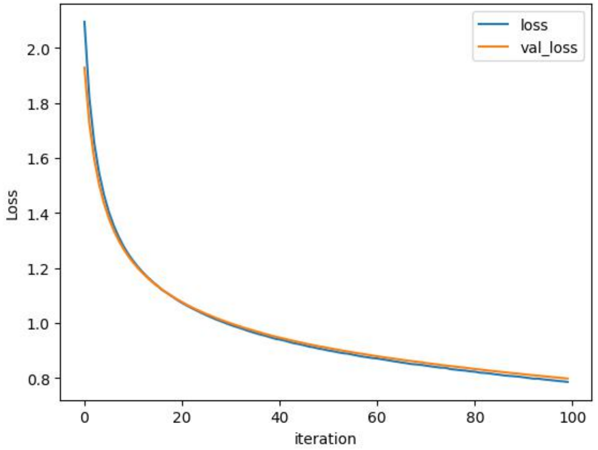
Train lần 1



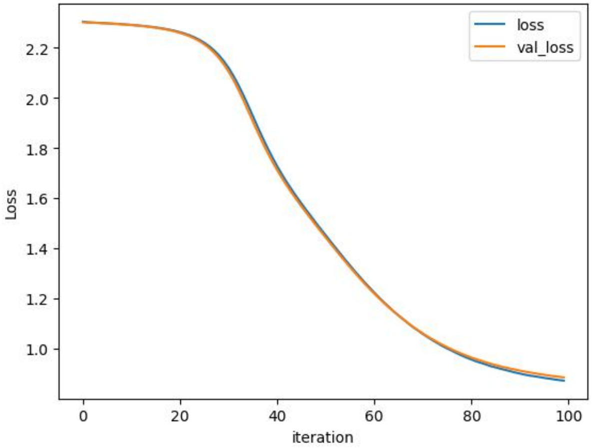
Train lần 2



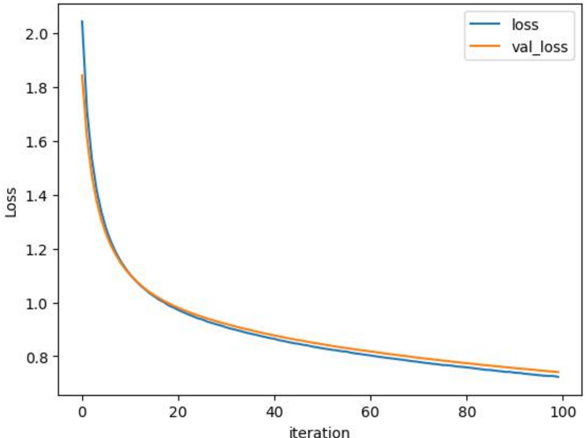
Train lần 3



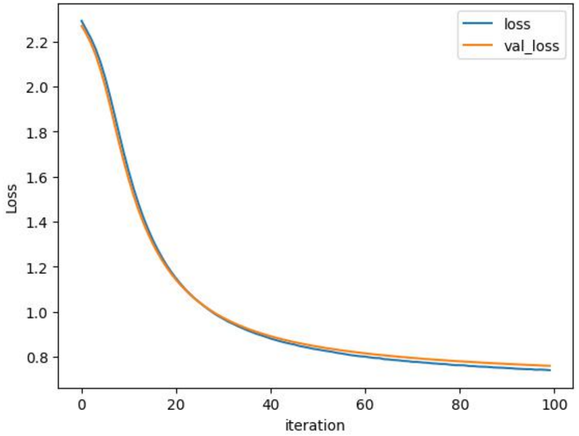
Train lần 4



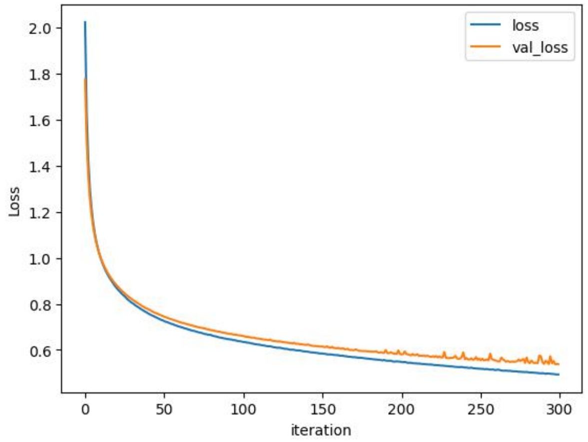
Train lần 5



Train lần 6



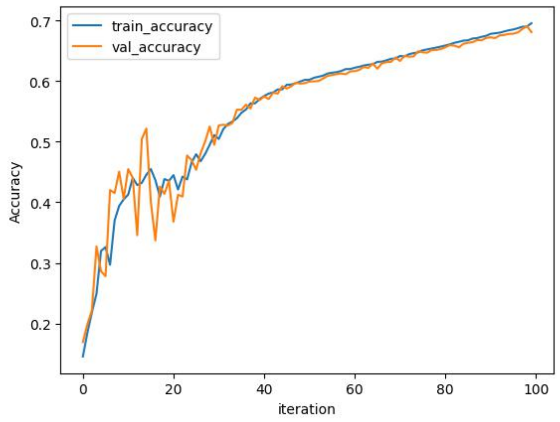
Train lần 7



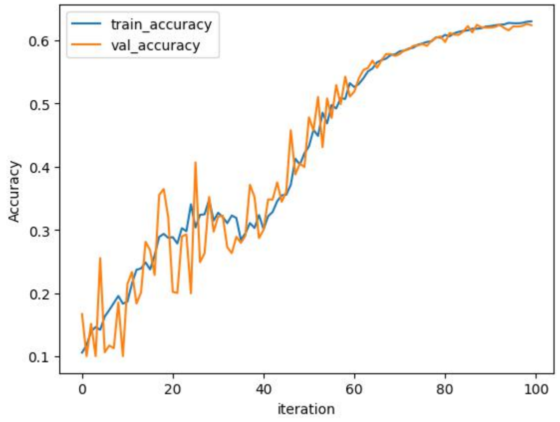
Fashion MNIST Vanishing Problem

- **Train Some Layer**

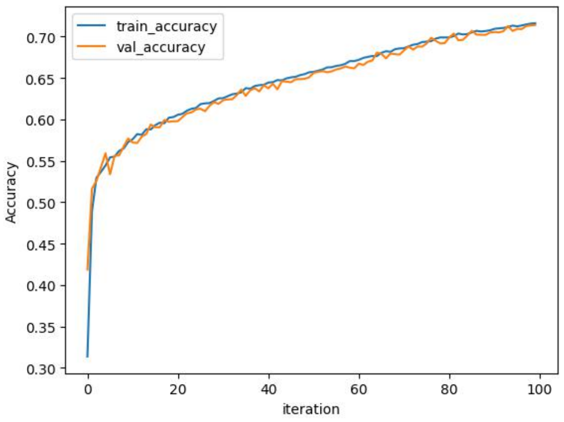
Train lần 1



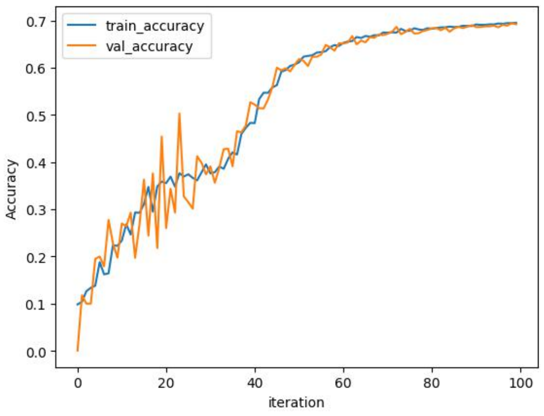
Train lần 2



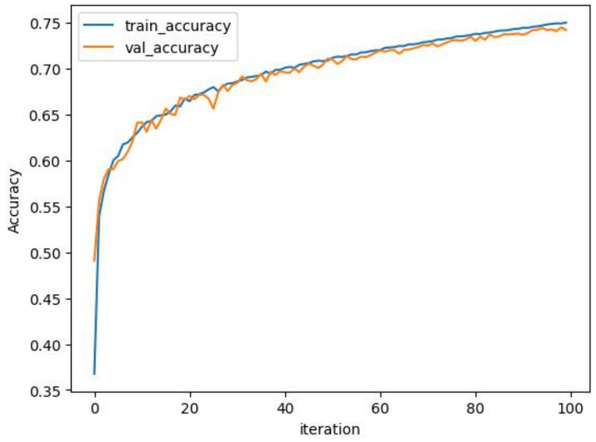
Train lần 3



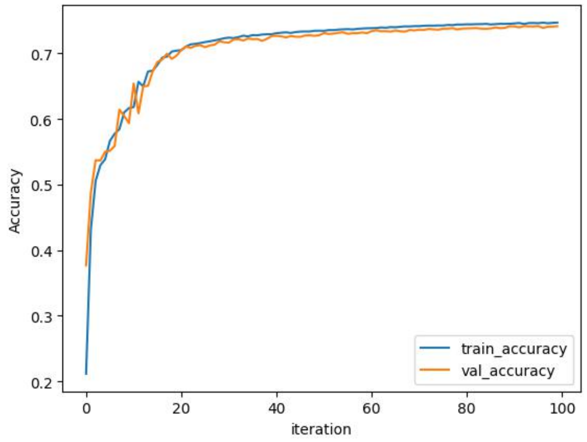
Train lần 4



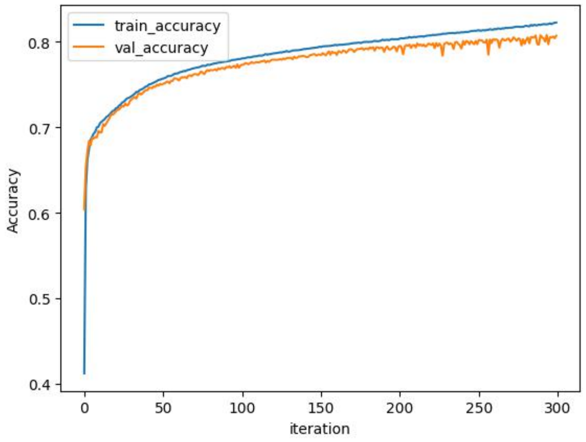
Train lần 5



Train lần 6



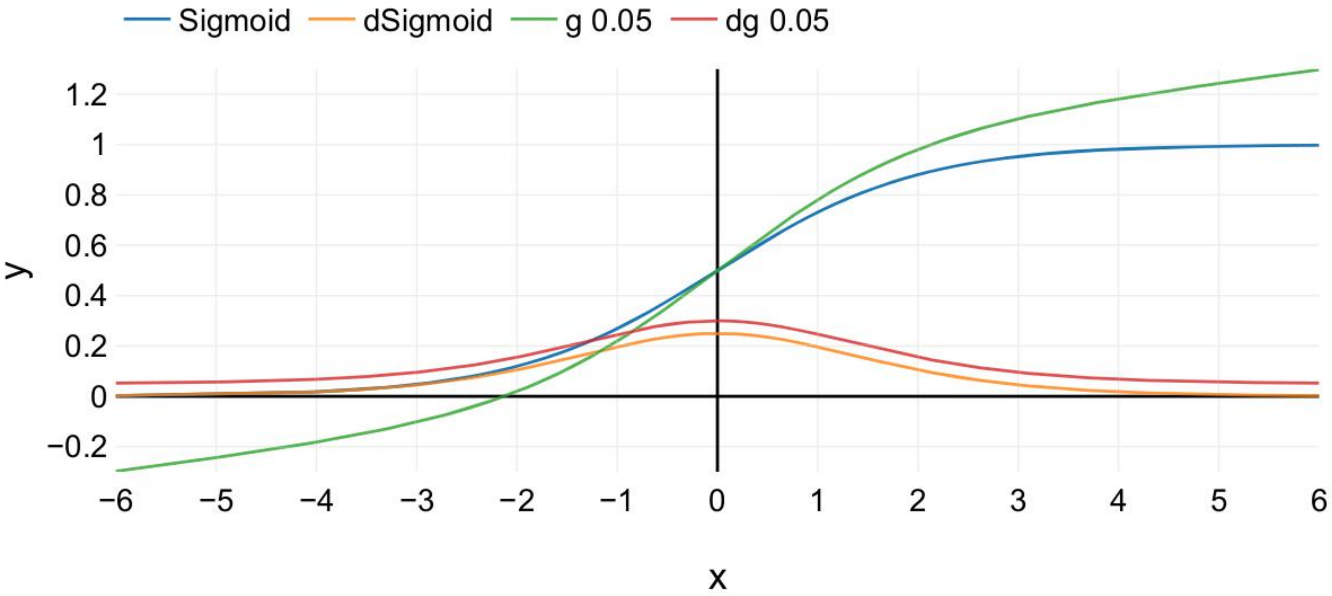
Train lần 7



Other Methods

Other Methods

- **A new approach for the vanishing gradient problem on sigmoid activation**



Activation functions with β = 0.05, and their derivative functions

$$\text{Sigmoid}(z) = \frac{1}{1+e^{-z}}$$

$$\frac{\mathrm{d}}{\mathrm{d}z}(\text{Sigmoid}(z)) = \text{Sigmoid}(z)(1-\text{Sigmoid}(z))$$

$$g(z) = \text{Sigmoid}(z) + \beta z$$

$$g'(z) = \text{Sigmoid}(z)(1+\text{Sigmoid}(z)) + \beta$$

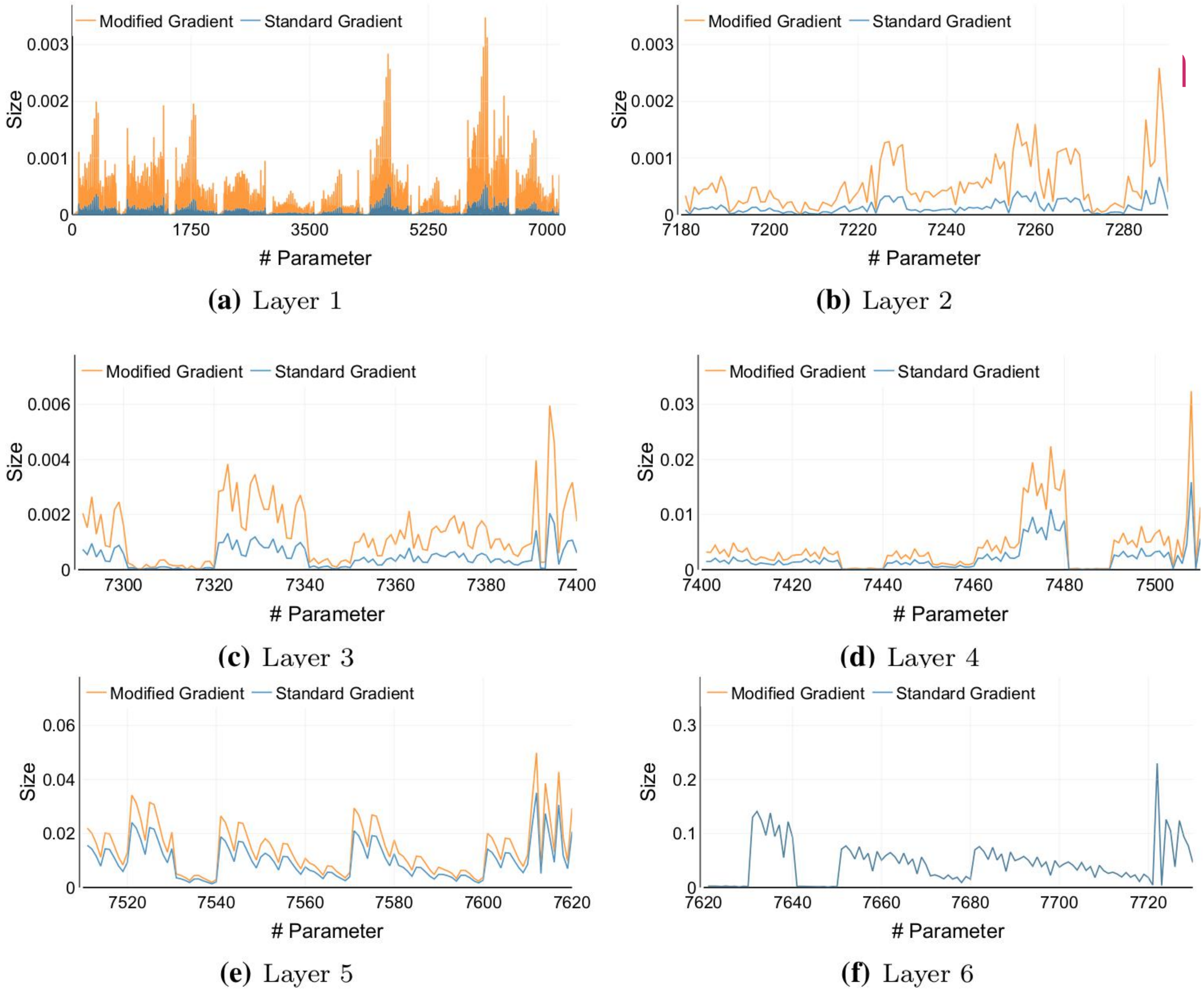
- It is very close to the original sigmoid function in the range [−1,1].
- It is a differentiable and unbounded function.
- The derivative of both functions differs by a constant equal to β. That is, the derivative of g is larger than that of the sigmoid in all R.
- When its argument tends to +∞ or −∞, the derivative is asymptotic to β.

Other Methods

- **A new approach for the vanishing gradient problem on sigmoid activation**

Algorithm 1 Backpropagation with modified derivative for sigmoid function

Input: x, y, θ: weights and bias, α: learning rate, β: modification parameter, f: function, f': derivative
Output: θ_{new}
Where: L is the number of layers,
for $epoch = 1, 2, \ldots, N$ **do**
    for $l = 1, 2, \ldots, L$ **do** (Compute Activations)
        if $l = 1$ **then**
            $a^{(0)} = x$
        end if
        $z^{(l)} = \theta^{(l)} a^{(l-1)}$
        $a^{(l)} = Sigmoid(z^{(l)})$
    end for
    for $l = L, L-1, \ldots, 1$ **do** (Compute δ)
        if $l = L$ **then**
            $\delta_{out}^{(l)} = -(y - a^{(l)}) \bullet f'_{out}(z^{(l)})$
        else
            $\delta_{\beta}^{(l)} = ((\theta^{(l)})^T \delta_{\beta}^{(l+1)}) \bullet (Sigmoid'(z^{(l)}) + \beta)$
        end if
    end for
    for $l = L, L-1, \ldots, 1$ **do** (Compute gradient)
        $\nabla_{\theta^{(l)},\beta} J(\theta; x, y) = \delta_{\beta}^{(l)} (a^{(l-1)})^T$
    end for
    for $l = L, L-1, \ldots, 1$ **do** (Update Parameters)
        $\theta_{new}^{(l)} = \theta_{old}^{(l)} - \alpha \nabla_{\theta^{(l)},\beta} J(\theta; x, y)$
    end for
end for



- **A new approach for the vanishing gradient problem on sigmoid activation**

Other Methods

Other Methods

- **Vanishing Gradient Analysis in Stochastic Diagonal Approximate Greatest Descent Optimization**

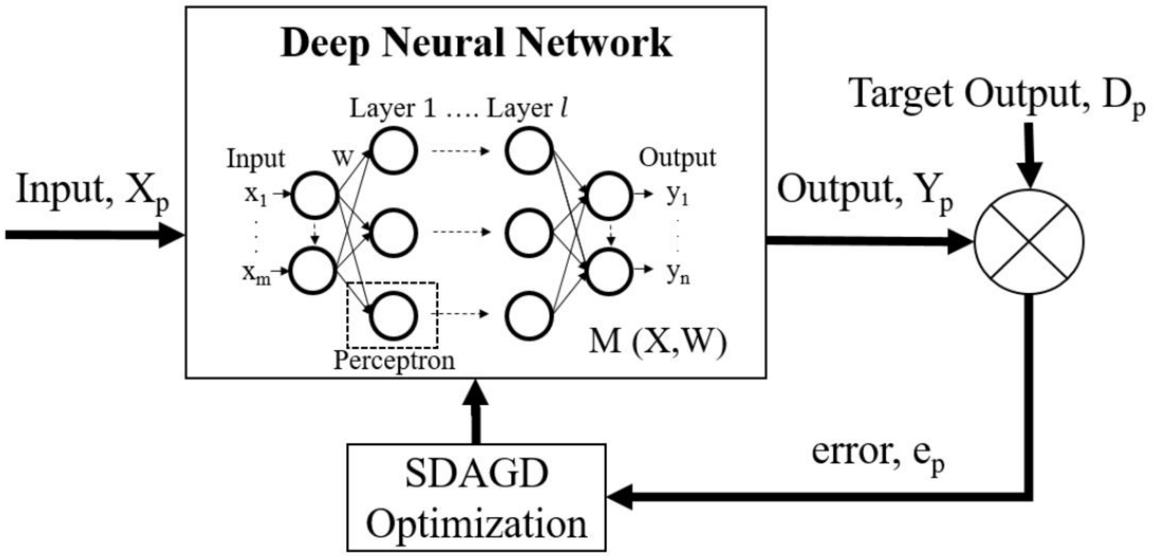


Fig. 1. Block diagram of deep learning neural networks with the proposed optimization method – Stochastic Diagonal Approximate Greatest Descent.

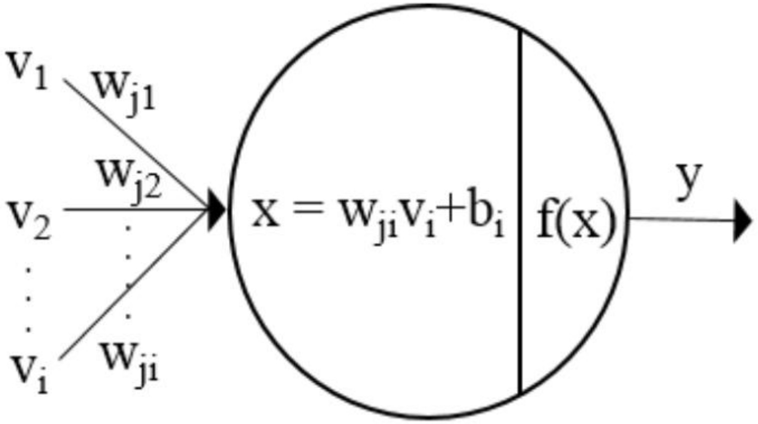


Fig. 2. Basic operation in a perceptron.

Other Methods

- **Vanishing Gradient Analysis in Stochastic Diagonal Approximate Greatest Descent Optimization**

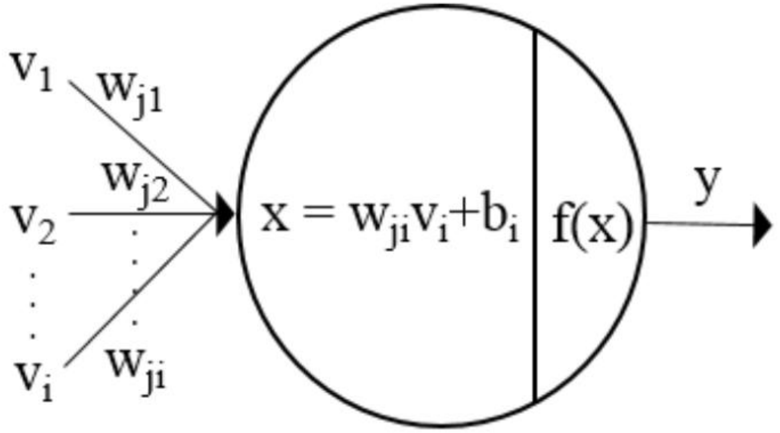


Fig. 2. Basic operation in a perceptron.

$$W_{k+1} = W_k + \eta\, g(W_k),$$

$$\boxed{W_{k+1} = W_k + [\mu_k J + H(W_k)]^{-1} g(W_k)}$$

where $\mu_k = \frac{\|g(W_k)\|}{R_k}$ is the relative step length

J is all-ones matrix.

$H(W_k)$ is the truncated Hessian matrix and R_k

R_k is the radius constant

Other Methods

- **Vanishing Gradient Analysis in Stochastic Diagonal Approximate Greatest Descent Optimization**

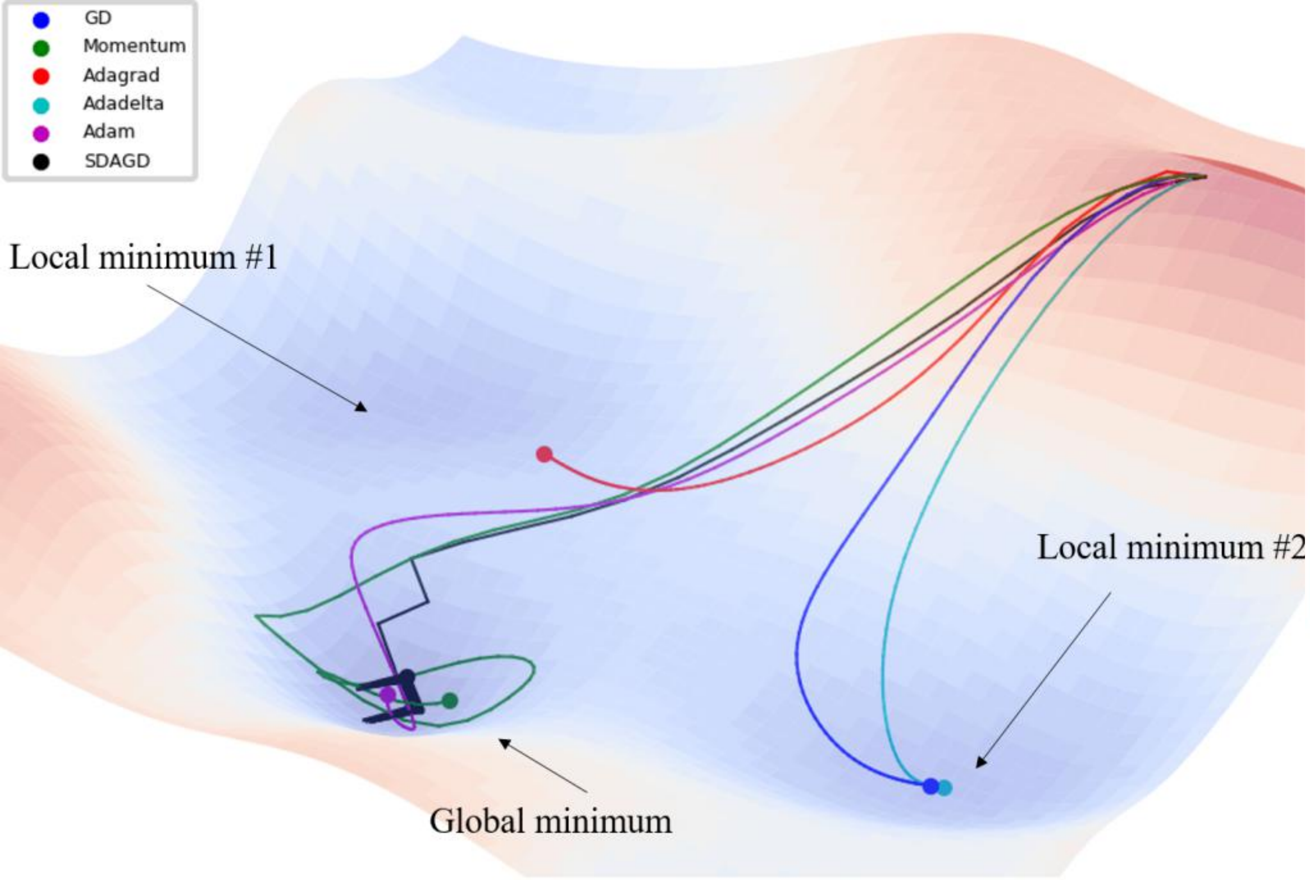


Fig. 3. A hilly error surface with two local minimums and one global minimum.

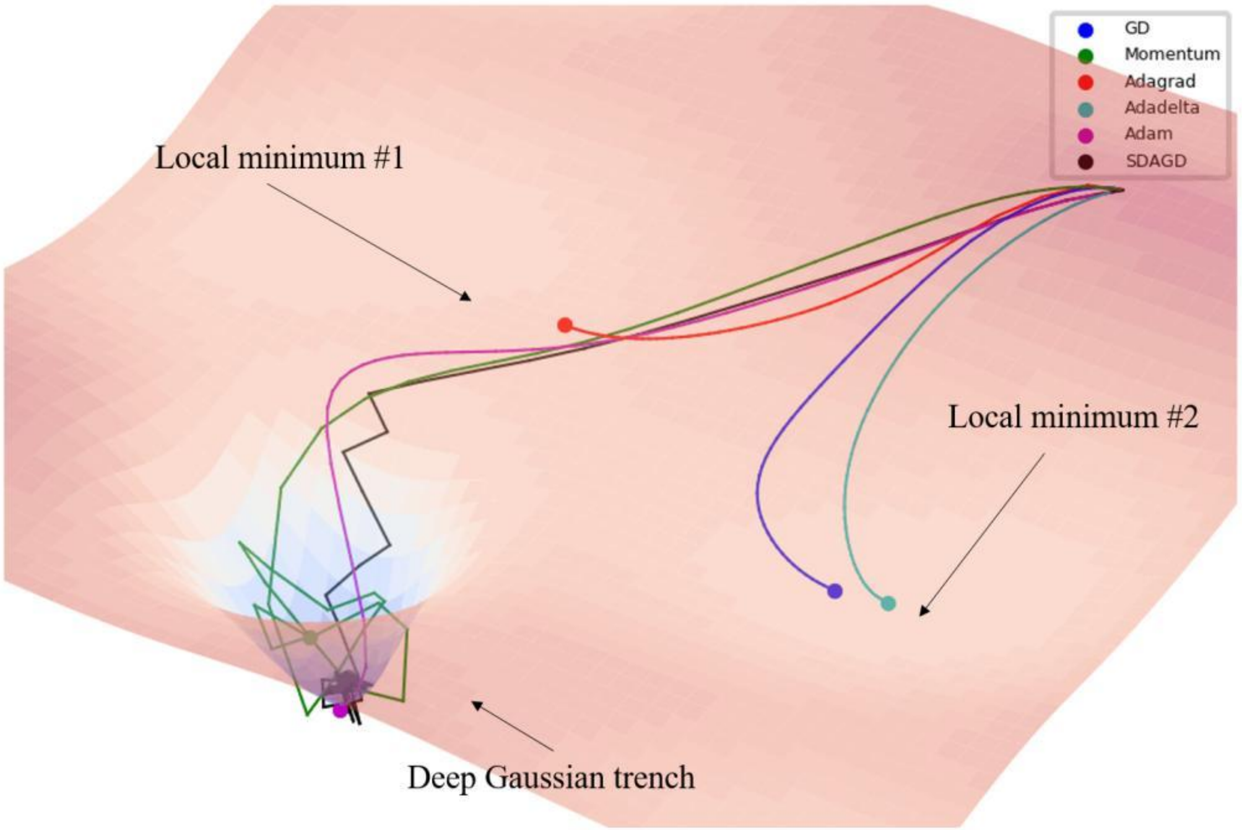


Fig. 4. A deep Gaussian trench to simulate drastic gradient changes.