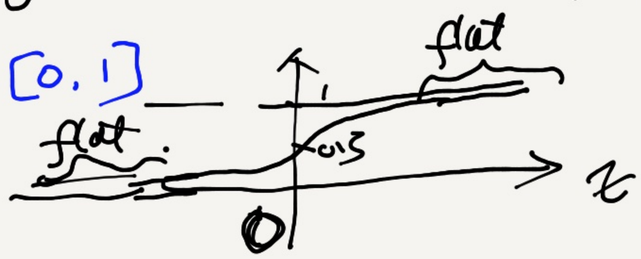
Lecture 13.

1. Vanishing gradient problem

1) Assume a multi-layer NN, with sigmoid activation functions. (all nodes)

$$\sigma(x) = \frac{1}{1+e^{-x}} \in [0,1]$$

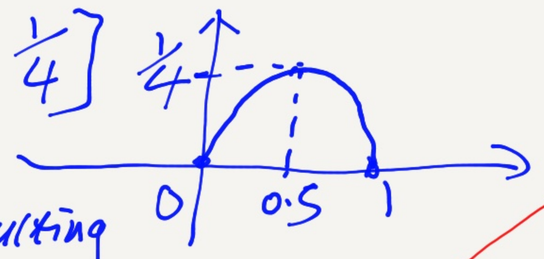


① If x is too large or too small, $\sigma(x)$ will saturate (0 or 1)

and $\sigma'(x)$ will approach 0.

$$\nabla_{w_{kj}} J = \delta_k \cdot h_j = (o_k - y_k)\cdot \sigma' \cdot h_j \longrightarrow 0 \quad \text{if } \sigma' \to 0$$

② If we use the BP through σ reduce the gradient by a factor of at least 4. $\sigma' = \sigma(1-\sigma) \in [0, \frac{1}{4}]$



③ Propagating through several layers, the resulting gradients become very small.

$$\nabla_{w_{ij}} J = \delta_j \cdot x_i = \sum_{k=1}^{c} (\delta_k w_{kj}) \cdot \sigma' \cdot x_i$$

w_{ij}: Weights between input and hidden layer

activation of hidden layer

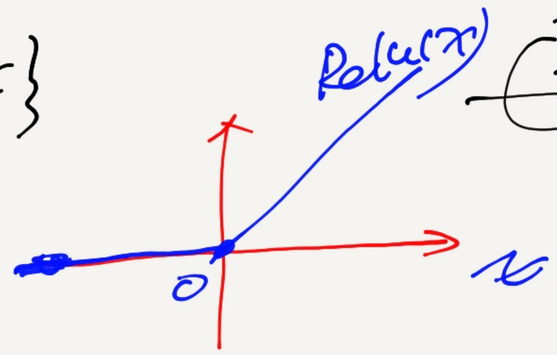
GD/SGD: $w^{i+1} = w^i - \underbrace{\varepsilon \cdot \nabla_{w^i} L}_{\to 0} \to w^{i+1} = w^i$

no update

2) solve the issue.

① use different activation function

$$ReLU(x) = \max\{0, x\}$$

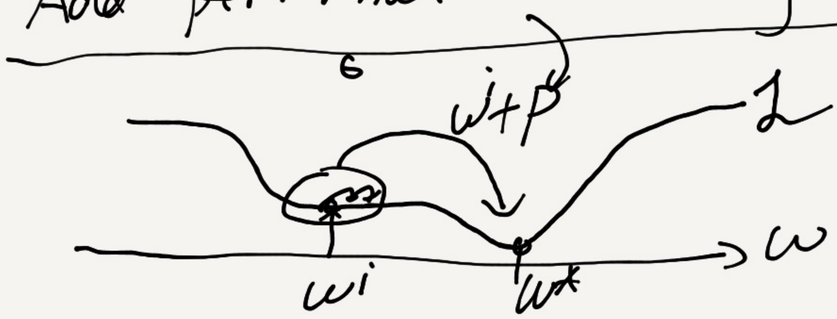


$$\nabla ReLU(x) = \begin{cases} 1, x > 0 \\ 0 \; x \le 0 \end{cases}$$

② Add momentum term during the training

$\nabla L \to 0$ —Add momentum.→ $\nabla L + M$

③ Add perturbations during training



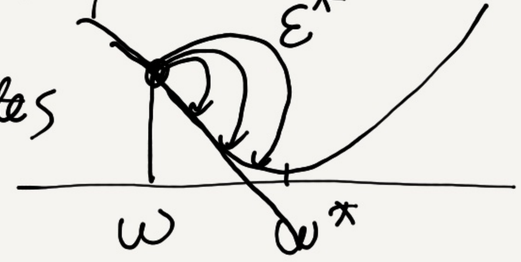
$$w^{i+1} = w^i - (\varepsilon \cdot \nabla_w L + M)$$

2. Multiple factors control the learning process

$$w^{i+1} = w^i - \varepsilon \cdot \nabla_{w^i} L$$

1) learning rate : { too small: training converges slowly.
too large: not coverge.

① use another algorithm to identify a 'good' learning rate.
- line search. → search an optimal ε^*



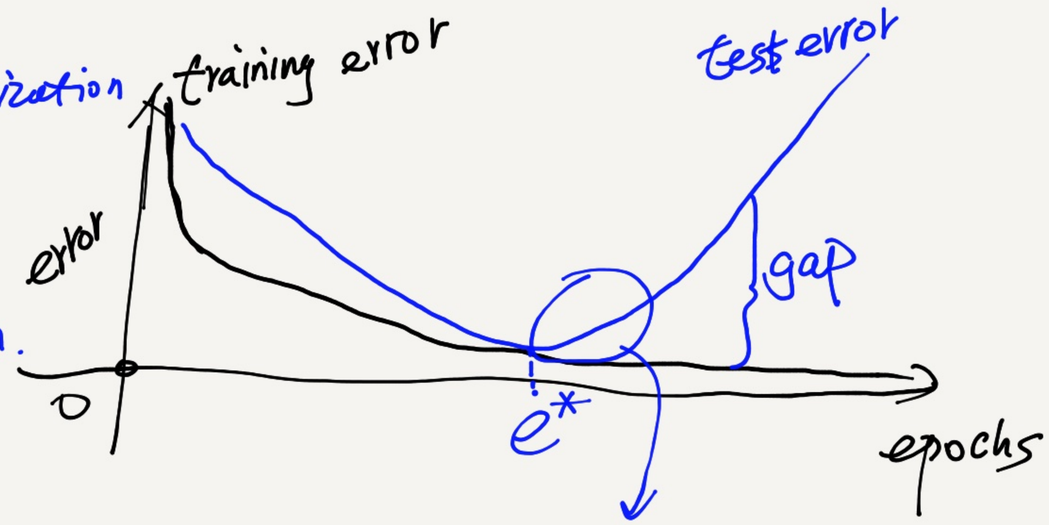
- Try different learning rates and choose the best.

② Determine the learning rate adaptively. → Adam

2) ∇L (avoid the vanishing gradient issue)

3. Early stopping and evaluation.

Excessive training leads to poor generalization (test perform.)



① stop training at the right epoch.

If the test/validation performance start to decrease, we get the signal to stop the training.

→ If we involve test set into the training (test curve), we will leak our test data (model observed the test data during training)

↓ solve this issue by splitting the whole datasets into 3 subsets: training, validation, test sets