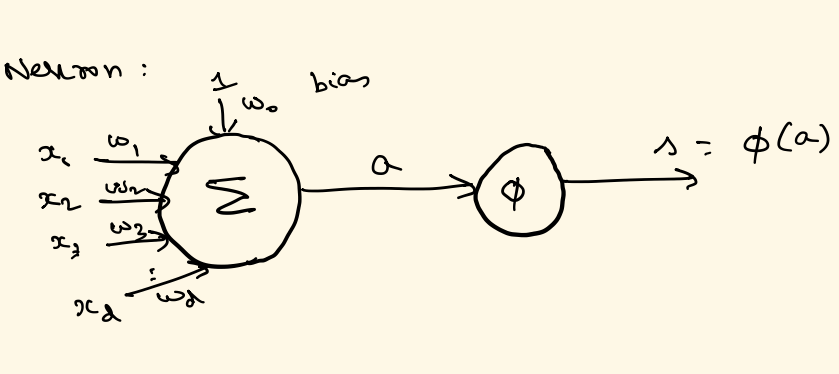
MLFFNNs

Monday, 29 January 2024 08:46

29/01/2024

Feedforward Neural Networks:

Artificial Neuron:



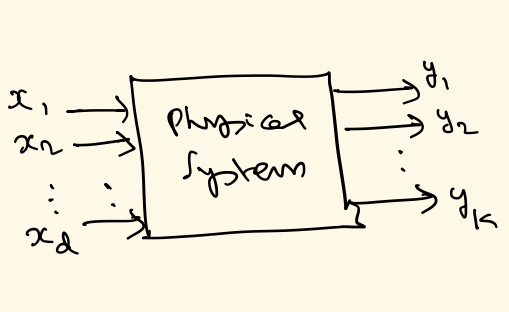
a : Activation value

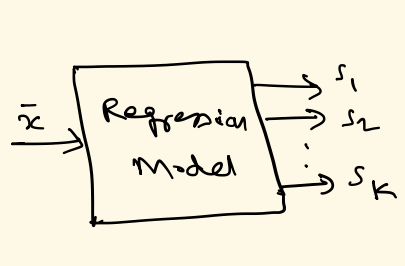
$$a = \sum_{i=1}^{d} w_i x_i + w_0$$

Φ : Activation fn

$$s = \Phi(a)$$

Regression and Classification Tasks





$$y_k = f_k(\bar{x})$$

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

$d \times 1$

$\hat{f}_k(\bar{x}) \approx f_k(\bar{x})$

Model output — Desired output

$\bar{x} \rightarrow y_k$

$\bar{x} \rightarrow s_k$

$y_k \in \mathbb{R}$

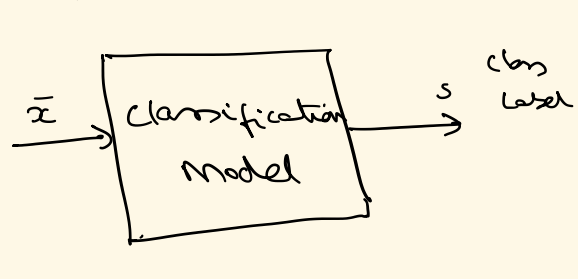
$s_k \in \mathbb{R}$

Classification:

Example $\bar{x}$ — class label y

No. of class labels: K

Set of class labels: $\{y_1, y_2, \ldots, y_K\}$

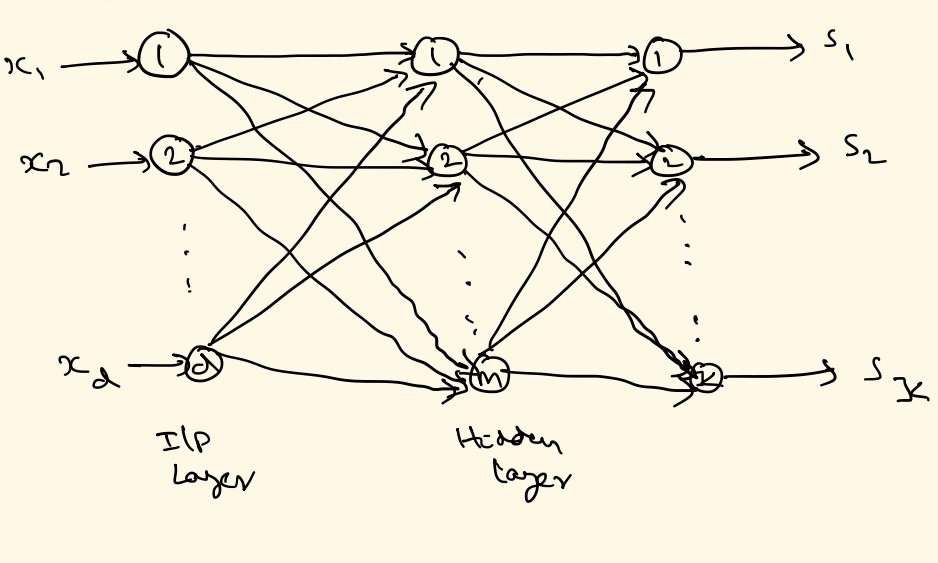


s : Discrete variable

No. of layers of neurons connected in a feedforward manner

Visible layers
- Input layer : $\bar{x}$ as input — d inputs
- Output layer : — K outputs

one or more hidden layers



Weight parameters associated with connections

w_{ij} : Weight of connection from node i in I/P layer to node j in hidden layer

w_{jk} : Weight of connection from node j in hidden layer to node k in output layer

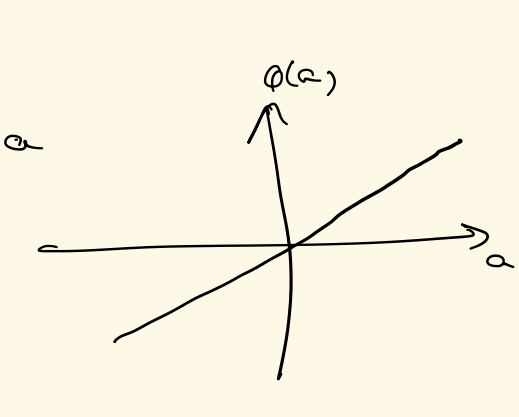
$i : 0, 1, \ldots d$

$j : 0, 1, \ldots M$

$k : 1, 2, \ldots, K$

Linear activation fn:

$$\phi(a) = a$$



Sigmoidal activation fn:

Logistic Sigmoid fn:

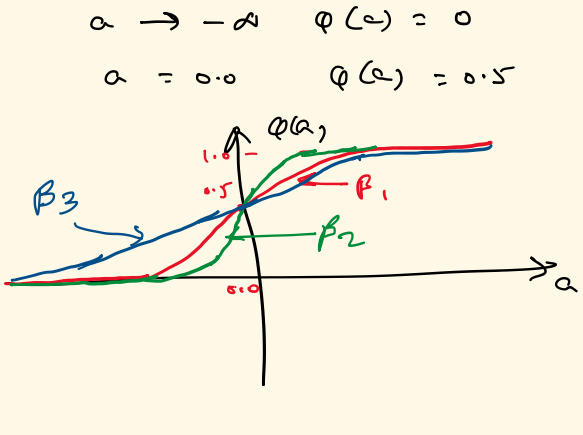
$$\phi(a) = \frac{1}{1+e^{-\beta a}}$$

β : Slope parameter

$a \rightarrow \infty \quad \phi(a) = 1$

$a \rightarrow -\infty \quad \phi(a) = 0$

$a = 0.0 \quad \phi(a) = 0.5$



$\beta_2 > \beta_1$

$\beta_3 < \beta_1$

$$\phi(a) = \frac{1}{1+e^{-\beta a}}$$

$$\frac{d\phi(a)}{da} = -\frac{1}{(1+e^{-\beta a})^2}\frac{d}{da}(1+e^{-\beta a})$$

$$= -\frac{1}{(1+e^{-\beta a})^2} \cdot -\beta \cdot e^{-\beta a}$$

$$= \beta \cdot \underbrace{\frac{1}{1+e^{-\beta a}}} \cdot \underbrace{\frac{e^{-\beta a}}{1+e^{-\beta a}}}$$

$$\frac{d\phi(a)}{da} = \beta \cdot \phi(a)(1-\phi(a))$$

$$\phi(a) = \frac{c}{1+e^{-\beta a}}$$

$$\frac{d\phi(a)}{da} = \frac{\beta}{c}\phi(a) \cdot (c - \phi(a))$$

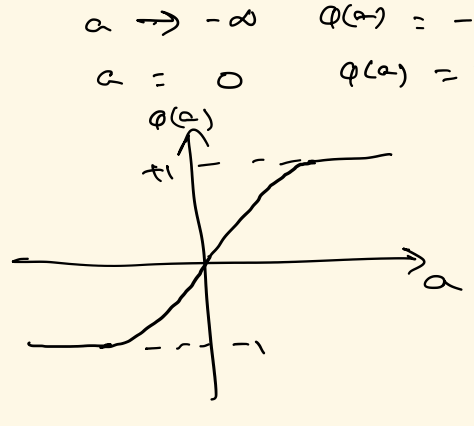
Hyperbolic Tangent Function:

$$\phi(a) = \tanh(\beta a) = \frac{e^{\beta a} - e^{-\beta a}}{e^{\beta a} + e^{-\beta a}}$$

$a \rightarrow \infty \quad \phi(a) = +1$

$a \rightarrow -\infty \quad \phi(a) = -1$

$a = 0 \quad \phi(a) = 0$



$$\frac{d\phi(a)}{da} = \frac{d}{da}\tanh(\beta a)$$

$$= \beta\,\mathrm{sech}^2(\beta a)$$

$$= \beta \cdot (1 - \tanh^2(\beta a))$$

$$= \beta(1-\tanh(\beta a))(1+\tanh(\beta a))$$

$$\frac{d\phi(a)}{da} = \beta(1-\phi(a))(1+\phi(a))$$

$$\phi(a) = c \cdot \tanh(\beta a)$$

$-c$ to $+c$

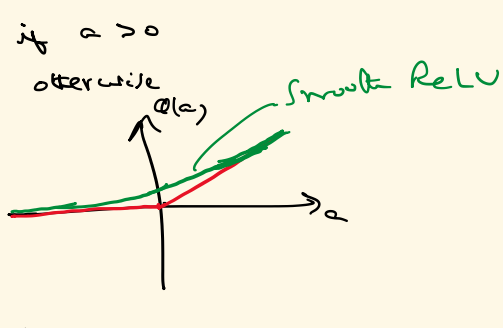
$$\frac{d\phi(a)}{da} = \frac{\beta}{c}(c-\phi(a))(c+\phi(a))$$

Properties of activation fn:
- Nonlinear
- continuous (Differentiable)
- Bounded
- Monotonically non-decreasing

Universal Approximation Theorem.

Rectified Linear Unit (ReLU)

$$\phi(a) = a, \text{ if } a > 0$$
$$= 0, \text{ otherwise}$$



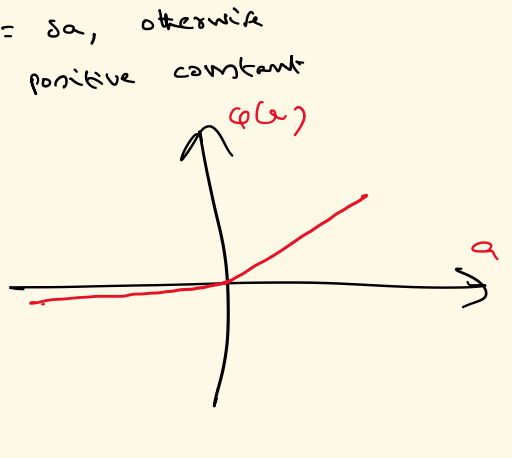
Smooth ReLU (Softplus)

$$\phi(a) = \log(1+e^{a})$$

Leaky ReLU

$$\phi(a) = a, \text{ if } a > 0$$
$$= \delta a, \text{ otherwise}$$

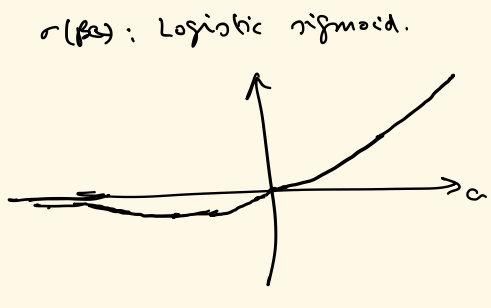
δ is a positive constant



Swish function:

$$\phi(a) = a\,\sigma(\beta a) = a \cdot \frac{1}{1+e^{-\beta a}}$$

$\sigma(\beta a)$: Logistic sigmoid.



$$\frac{d\phi(a)}{da} = \frac{d}{da}\left(a \cdot \frac{1}{1+e^{-\beta a}}\right)$$

$$= \frac{1}{1+e^{-\beta a}} + a \cdot \frac{-1}{(1+e^{-\beta a})^2} \cdot -\beta \cdot e^{-\beta a}$$

$$= \frac{\phi(a)}{a} + a \cdot \beta \cdot \frac{1}{1+e^{-\beta a}} \cdot \frac{e^{-\beta a}}{1+e^{-\beta a}}$$

$$= \frac{\phi(a)}{a} + a \cdot \beta \cdot \frac{\phi(a)}{a} \cdot \left(1 - \frac{\phi(a)}{a}\right)$$

$$= \frac{\phi(a)}{a}\left[1 + a\beta \cdot \left(1 - \frac{\phi(a)}{a}\right)\right]$$

Softmax function:

K classes

$l = 1, 2, \ldots, K$

a_l : Activation value of lth node in o/p layer

$$\phi(a_l) = \frac{e^{a_l}}{\sum_{m=1}^{K} e^{a_m}}$$

$$0.0 \leq \phi(a_l) \leq 1.0$$

$$\sum_{l=1}^{K} \phi(a_l) = 1.0$$

$$\frac{d\phi(a_l)}{da_l} = \frac{d}{da_l}\left(\frac{e^{a_l}}{\sum_{m=1}^{K} e^{a_m}}\right)$$

$$= e^{a_l} \cdot \frac{1}{\sum_{m=1}^{K} e^{a_m}} + e^{a_l} \cdot \frac{-1}{\left(\sum_{m=1}^{K} e^{a_m}\right)^2} e^{a_l}$$

$$= \phi(a_l) - \phi(a_l) \cdot \phi(a_l)$$

$$\frac{d\phi(a_l)}{da_l} = \phi(a_l)(1-\phi(a_l))$$

Function approximation task:

Activation fn in output layer

Range of output variable	Activation fn
0.0 to 1.0	Logistic
0.0 to c	Scaled logistic
−1.0 to 1.0	Tanh
−c to +c	Scaled Tanh
0 to ∞	ReLU
−∞ to ∞	Linear

Classification task:

Example $\bar{x}_n$: class label from a set of K class labels

$\bar{x}_n$: lth class

One-hot representation of desired output vector

$\bar{t}_n$: Desired output vector. $K \times 1$

$t_{nl} = 1$

$t_{nm} = 0, \; m \neq l$

1-of-K representation

Sum-of-squared errors

$$\sum_{m=1}^{K}(t_{nm} - y_{nm})^2$$

$t_{nl}=1, t_{nm}=0$: Logistic sigmoid fn

$t_{nl} = +1$

$t_{nm} = -1, \; m \neq l$

tanh fn.

Cross-entropy error fn

Softmax fn

Supervised Learning:

Training dataset : $\mathcal{D} = \{\bar{x}_n, \bar{t}_n\}_{n=1}^{N}$

$\bar{x}_n \in \mathbb{R}^d$

Regression $\bar{t}_n \in \mathbb{R}^K$

classification $\bar{t}_n \in \{1, 0\}^K$ Logistic Softmax

$\bar{t}_n \in \{+1, -1\}^K$ Tanh

Error correction Learning

Error Backpropagation Method (BP)

Sum-of-squared errors:

$\bar{x}_n \quad \bar{t}_n \quad \bar{y}_n$ Actual output of model

$\bar{y}_n \in \mathbb{R}^K$

Instantaneous Error

$$\tilde{E}_n = \frac{1}{2}\|\bar{t}_n - \bar{y}_n\|^2 = \frac{1}{2}\sum_{l=1}^{K}(t_{nl} - y_{nl})^2$$

Cross-Entropy error:

Discrete Random variable : X

$P(X), \; Q(X)$

$$CE = \underset{P(X)}{E}[-\ln Q(X)]$$

$P(X) : \bar{t}_n$

$Q(X) : \bar{y}_n$

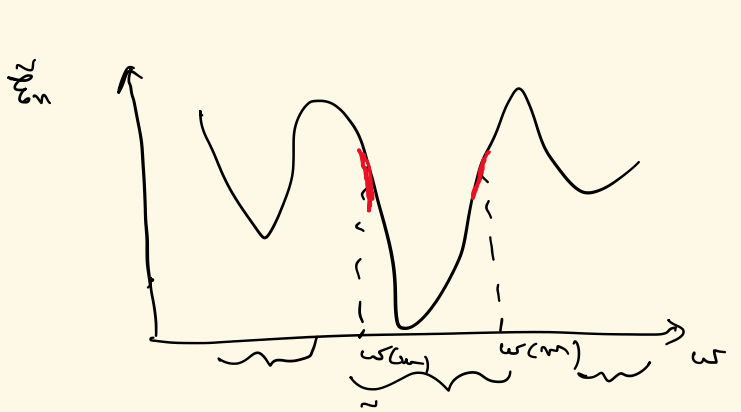
$$\tilde{E}_n = -\sum_{l=1}^{K} t_{nl} \ln y_{nl}$$

$t_{nl} = 1$, if $\bar{x}_n$ belongs to lth class

$= 0$, otherwise

$$\tilde{E}_n = -\ln y_{nl}$$

Gradient Descent Method



$$\Delta w = -\eta \left.\frac{\partial \tilde{E}_n}{\partial w}\right|_{w=w(m)}$$

η : Learning rate parameter

$0.0 < \eta \leq 1.0$

Pattern Mode:

Step m : Pick an example $\bar{x}_n$

$$w(m+1) = w(m) + \Delta w(m)$$

Delta Rule

$$w(m+1) = w(m) - \eta \left.\frac{\partial \tilde{E}_n}{\partial w}\right|_{w=w(m)}$$

Epoch: Weight update for all N examples

N updates.

Stochastic Gradient Descent (SGD) method

Batch Mode:

$$w(m+1) = w(m) - \eta \left.\frac{\partial E_{avg}}{\partial w}\right|_{w=w(m)}$$

$$E_{avg} = \frac{1}{N}\sum_{n=1}^{N}\tilde{E}_n$$

$$\frac{\partial E_{avg}}{\partial w} = \frac{1}{N}\sum_{n=1}^{N}\left.\frac{\partial \tilde{E}_n}{\partial w}\right|_{w=w(m)}$$

Epoch: 1 update.

Mini-Batch Mode:

$\mathcal{D}$: Form subsets

Size of subset: M

$\mathcal{D}_b, \quad b = 1, 2, \ldots, B$

$$B = \left\lceil \frac{N}{M} \right\rceil$$

For $\mathcal{D}_b$,

$$w(m+1) = w(m) - \eta \left.\frac{\partial E^b_{avg}}{\partial w}\right|_{w=w(m)}$$

$$\frac{\partial E^b_{avg}}{\partial w} = \frac{1}{M}\sum_{\bar{x}_n \in \mathcal{D}_b}\left.\frac{\partial \tilde{E}_n}{\partial w}\right|_{w=w(m)}$$

Epoch: B updates

Training Process:

1. Initialize all parameters of the model (weights, biases)
2. Perform weight update(s) in an epoch using all the N examples in $\mathcal{D}$
3. Repeat Step 2 until a terminating criterion is satisfied

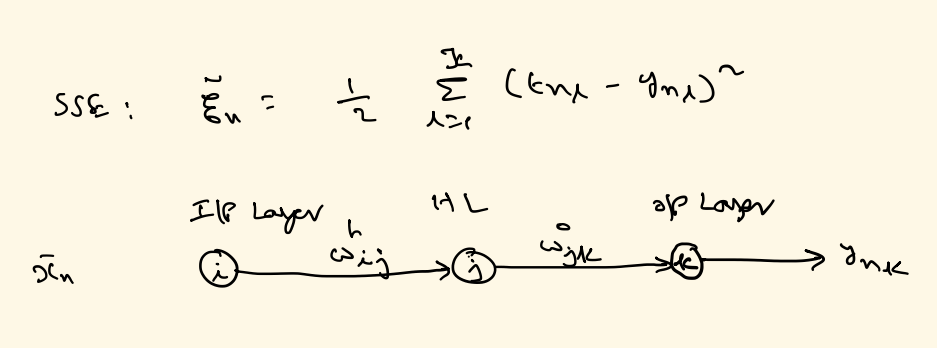
$$\frac{\partial \tilde{E}_n}{\partial w}$$

$\tilde{E}_n$: Instantaneous error

SSE : $\tilde{E}_n = \frac{1}{2}\sum_{l=1}^{K}(t_{nl} - y_{nl})^2$

CE : $\tilde{E}_n = -\ln y_{nl}$ — $\bar{x}_n$ belongs to class l

SSE : $\tilde{E}_n = \frac{1}{2}\sum_{l=1}^{K}(t_{nl} - y_{nl})^2$



Forward pass of computation:

Compute the outputs of all nodes

$$s_i = x_{ni}$$

$$s^h_j = \phi^h(a^h_j)$$

$$y_{nk} = s^o_k = \phi^o(a^o_k)$$

$$a^h_j = \sum_{i=1}^{d} w^h_{ij} s_i + w_{0j}$$

$$a^o_k = \sum_{j=1}^{J} w^o_{jk} s^h_j + w_{0k}$$

Backward Pass:

$$\frac{\partial \tilde{E}_n}{\partial w^o_{jk}} = \frac{\partial}{\partial w^o_{jk}}\frac{1}{2}\sum_{l=1}^{K}\underbrace{(t_{nl} - y_{nl})^2}$$

$$= -(t_{nk} - y_{nk})\frac{\partial y_{nk}}{\partial w^o_{jk}}$$

$$= -(t_{nk} - y_{nk})\frac{\partial \phi^o(a^o_k)}{\partial w^o_{jk}}$$

$$= -(t_{nk} - y_{nk})\underbrace{\frac{d\phi^o(a^o_k)}{da^o_k}} \cdot \underbrace{\frac{\partial a^o_k}{\partial w^o_{jk}}}$$

$$\frac{\partial \tilde{E}_n}{\partial w^o_{jk}} = \underbrace{-(t_{nk} - y_{nk})\frac{d\phi^o(a^o_k)}{da^o_k}}_{\frac{\partial \tilde{E}_n}{\partial a^o_k}} \cdot s^h_j$$

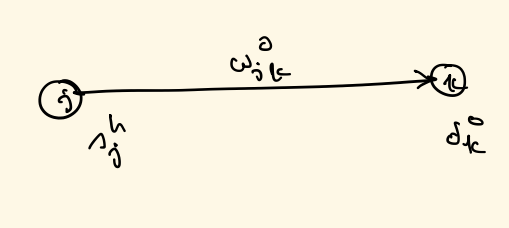
Local gradient δ^o_k

$$\delta^o_k = -\frac{\partial \tilde{E}_n}{\partial a^o_k}$$

$$\Delta w^o_{jk} = -\eta\frac{\partial \tilde{E}_n}{\partial w^o_{jk}}$$

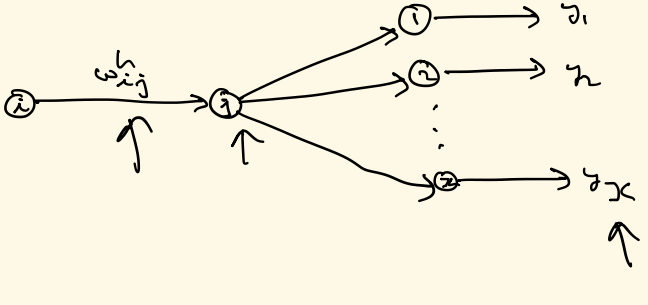
$$= \eta(t_{nk} - y_{nk}) \cdot \frac{d\phi^o(a^o_k)}{da^o_k} \cdot s^h_j$$

$$\boxed{\Delta w^o_{jk} = \eta\,\delta^o_k\,s^h_j}$$



$$\Delta w^h_{ij} = -\frac{\partial \tilde{E}_n}{\partial w^h_{ij}}$$

$$\frac{\partial \tilde{E}_n}{\partial w^h_{ij}} = \frac{\partial}{\partial w^h_{ij}}\frac{1}{2}\sum_{l=1}^{K}(t_{nl} - y_{nl})^2$$



$$\frac{\partial \tilde{E}_n}{\partial w^h_{ij}} = \frac{1}{2}\sum_{l=1}^{K} -2(t_{nl} - y_{nl}) \cdot \frac{\partial y_{nl}}{\partial w^h_{ij}}$$

$$= \sum_{l=1}^{K}\underbrace{-(t_{nl} - y_{nl}) \cdot \frac{d\phi^o(a^o_l)}{da^o_l}} \cdot \frac{\partial a^o_l}{\partial w^h_{ij}}$$

$$= \sum_{l=1}^{K} -\delta^o_l \frac{\partial}{\partial w^h_{ij}}\underbrace{\sum_{m=1}^{J} w^o_{ml} s^h_m + w_{0l}}$$

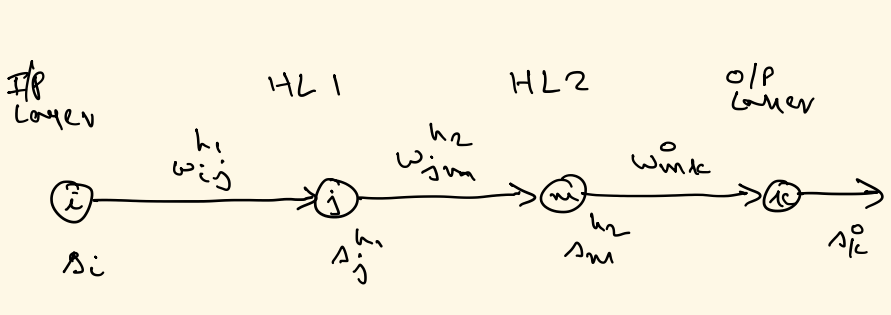
$$= \underbrace{\sum_{l=1}^{K} -\delta^o_l\,w^o_{jl} \cdot \frac{d\phi^h(a^h_j)}{da^h_j}}_{-\delta^h_j} \cdot \frac{\partial a^h_j}{\partial w^h_{ij}}$$

s_i

x_i

$$\Delta w^h_{ij} = \eta\,\delta^h_j\,s_i$$

$$\delta^h_j = \left(\sum_{l=1}^{K} w^o_{jl}\,\delta^o_l\right)\frac{d\phi^h(a^h_j)}{da^h_j}$$



$$\Delta w^o_{mk} = \eta\,\delta^o_k\,s^{h_2}_m$$

$$\delta^o_k = -\frac{\partial \tilde{E}_n}{\partial a^o_k} = (t_{nk} - y_{nk})\frac{d\phi^o(a^o_k)}{da^o_k}$$

$$\Delta w^{h_2}_{jm} = \eta\,\delta^{h_2}_m\,s^{h_1}_j$$

$$\delta^{h_2}_m = \left(\sum_{k=1}^{K} w^o_{mk}\,\delta^o_k\right)\frac{d\phi^{h_2}(a^{h_2}_m)}{da^{h_2}_m}$$

$$\Delta w^{h_1}_{ij} = \eta\,\delta^{h_1}_j\,s_i$$

$$\delta^{h_1}_j = \left(\sum_{m=1}^{M} w^{h_2}_{jm}\,\delta^{h_2}_m\right)\frac{d\phi^{h_1}(a^{h_1}_j)}{da^{h_1}_j}$$

Vanishing and Exploding Gradients Problem.

Stopping Criteria:

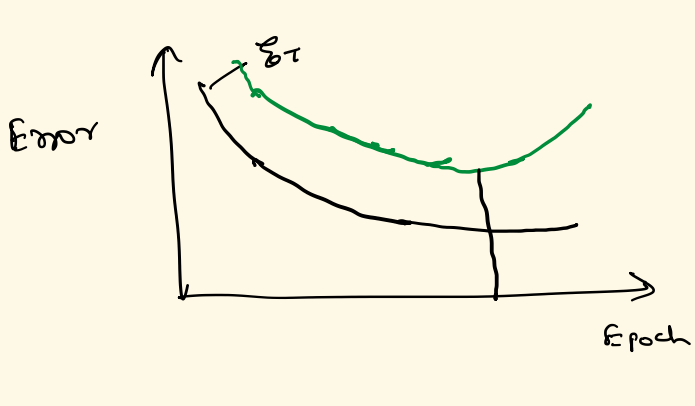
1. Change in E_{avg} from one epoch to the next epoch is below a threshold

   Hyperparameter

2. Early stopping

   Training error E_T

   Validation error E_V



3. No. of Epochs

$$\mathcal{D} = \{\bar{x}_n, \bar{t}_n\}_{n=1}^{N}$$

$d \times 1$

$\bar{x}_n \in \mathbb{R}^d$

Feature normalization:
(Input normalization)

Min-Max Normalization

$x_1, x_2, \ldots x_i, \ldots x_d$

$x_i : \quad x_{i\,max} \quad x_{i\,min}$

Normalized x_i

$$\hat{x}_i = \frac{x_i - x_{i\,min}}{x_{i\,max} - x_{i\,min}} \qquad 0.0 \leq \hat{x}_i \leq 1.0$$

Mean-Variance Normalization (standardization)

$x_i \quad \mu_i \quad \sigma_i$

$$\hat{x}_i = \frac{x_i - \mu_i}{\sigma_i}$$

$$\hat{\mu}_i = \frac{1}{N}\sum_{n=1}^{N}\frac{x_{ni} - \mu_i}{\sigma_i}$$

$$= \frac{\frac{1}{N}\sum_{n=1}^{N} x_{ni} - \mu_i}{\sigma_i}$$

$$= \frac{\mu_i - \mu_i}{\sigma_i}$$

$$= 0$$

$$\hat{\sigma}_i^2 = E\left[\frac{(x_i - \mu_i)^2}{\sigma_i^2}\right]$$

$$= 1$$

$$-3\hat{\sigma}_i \approx \hat{x}_i \approx +3\hat{\sigma}_i$$

$$-3.0 \approx \hat{x}_i \approx +3.0$$

Initialization of parameters:

Biases are initialized to zero

weights:

$$w \sim N(\overset{\mu}{0}, \overset{\sigma}{\epsilon})$$

$$\epsilon = 10^{-2}$$

Fan-in of a neuron: M

No. of inputs to a neuron

Xavier Glorot

Xavier Initialization
Glorot initialization

$$w \sim N\left(0, \sqrt{\tfrac{1}{M}}\right)$$

Linear
Sigmoid
Softmax

$$w \sim U\left(-\sqrt{\tfrac{1}{M}}, \sqrt{\tfrac{1}{M}}\right)$$

Uniform distribution.

Kaiming He

He initialization : $w \sim N\left(0, \sqrt{\tfrac{2}{M}}\right)$

Cross-Entropy error fn

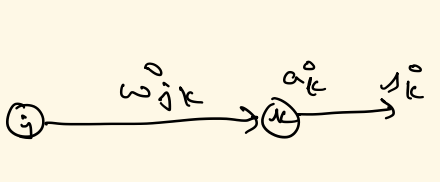
Softmax activation fn in the output layer

$\tilde{E}_n = -\ln y_k$ — $\bar{x}_n$ belongs to class k

$$\Delta w^o_{jk} = \eta(1 - s^o_k)s^h_j = \eta\,\delta^o_k\,s^h_j$$

$$s^o_k = \phi^o(a^o_k) = \frac{e^{a^o_k}}{\sum_{m=1}^{K} e^{a^o_m}}$$

$$\delta^o_k = (1 - s^o_k)$$



$m \neq k$

$$\Delta w^o_{jm} = -\eta\,s^o_m\,s^h_j = \eta\,\delta^o_m\,s^h_j$$

$$\delta^o_m = -s^o_m$$

$$\Delta w^h_{ij} = \eta\,\delta^h_j\,s_i$$

$$\delta^h_j = \left(\sum_{l=1}^{K} w^o_{jl}\,\delta^o_l\right)\frac{d\phi^h(a^h_j)}{da^h_j}$$

Hyperparameters:

- No. of hidden layers
- No. of nodes in each of the hidden layers
- β of sigmoid fn
- c : Scale parameter of sigmoid fn
- η : Learning rate
- Mode of learning
  - Pattern mode
  - Batch mode
  - Mini-batch mode
    - M: Mini-batch size

Delta Rule:

$$\Delta w = -\eta\frac{\partial E}{\partial w}$$