STAT 535

10/27/22

Lecture 9

Neural Networks

Qϕ, Q1 back

HW 3 posted

Lecture Notes III – Neural Networks

Marina Meilă
mmp@stat.washington.edu

Department of Statistics
University of Washington

October, 2020

Two-layer Neural Networks

Multi-layer neural networks

A zoo of multilayer networks

Reading HTF Ch.: 11.3 Neural networks, Murphy Ch.: (16.5 neural nets) and Dive Into Deep Learning 4.1-4.3

Two-layer Neural Networks

- The **activation function** (a term borrowed from neuroscience) is any continuous, bounded and strictly increasing function on $\mathbb{R}$. Almost universally, the activation function is the **logistic** (or **sigmoid**)

$$\phi(z) = \frac{1}{1+e^{-z}} \tag{1}$$

  because of its nice additional computational and statistical properties.
- We build a **two-layer neural network** in the following way:

Inputs	x_k	$k = 1 : d$
Bottom layer[1]	$z_j = \phi(w_j^T x)$	$j = 1 : m,\ w_j \in \mathbb{R}^n$
Top layer	$f = \phi(\beta^T z)$	$\beta \in \mathbb{R}^m$
Output	f	$\in [0, 1]$

  In other words, the neural network implements the function

$$f(x) = \sum_{j=1}^{m} \beta_j z_j = \sum_{j=1}^{m} \beta_j \phi(\sum_{k=1}^{m} w_{kj} x_k) \in (-\infty, \infty) \tag{2}$$

  Note that this is just a linear combination of logistic functions.

[1] In neural net terminology, each variable z_j is a **unit**, the bottom layer is **hidden**, while top one is **visible**, and the units in this layer are called hidden/visible units as well. Sometimes the inputs are called **input units**; imagine neurons or individual circuits in place of each x, y, z variable.

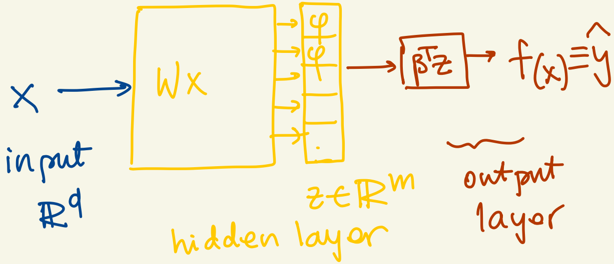
$X \in \mathbb{R}^d$

$y \in \mathcal{Y} \begin{cases} \to \mathbb{R} \\ \to \pm 1 \\ \to 1, 2, \ldots r \end{cases}$

output [layer]

$f(x) = \beta^T z(x)$ $\Rightarrow$ parameters $\beta \in \mathbb{R}^m$

$W = [W_{jk}]_{\substack{j=1:m \\ k=1:d}}$



$z_j(x) = \varphi(W_j x)$ — activation function

$z \in \mathbb{R}^m$

↳ hidden units ("variable")

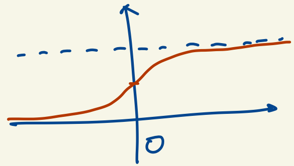
$\left(\begin{array}{l} x_k = \text{input unit} \\ f = \text{output unit} \end{array} \right)$

Hidden layer

$\varphi = \begin{cases} \text{sigmoid (logistic!)} \\ \text{ReLU} \end{cases}$

$\varphi(u) = \frac{1}{1+e^{-u}} = \frac{e^u}{1+e^u}$

bounded, smooth (Lipschitz)



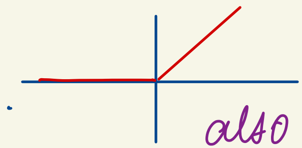
- $f(u)$ is L-Lipschitz $\Leftrightarrow$

$|f(u) - f(u')| \leq L \|u - u'\|$

$L > 0$ a parameter

$\varphi(u) = \begin{cases} u & \text{for } u \geq 0 \\ 0 & \text{for } u < 0 \end{cases}$

unbounded
non-smooth at 0
Lipschitz
$L = 1$



also called 'hinge'

(Rectified Linear Unit)

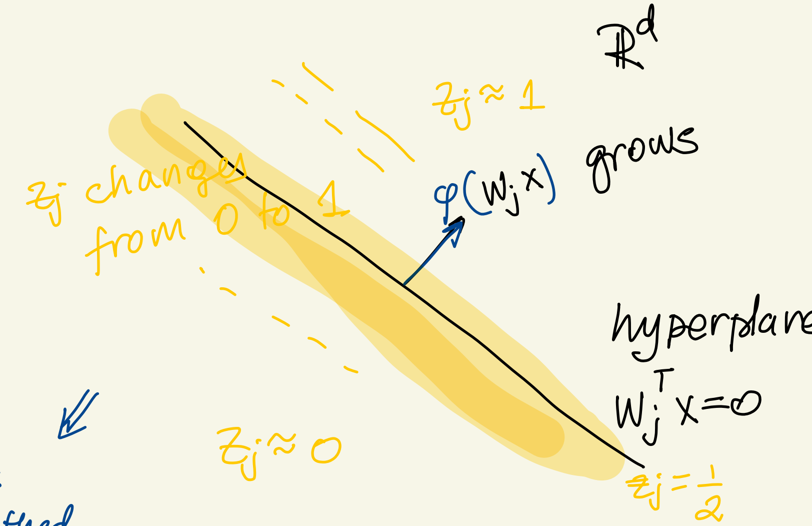
What does 2 layer NN represent?

φ = sigmoid

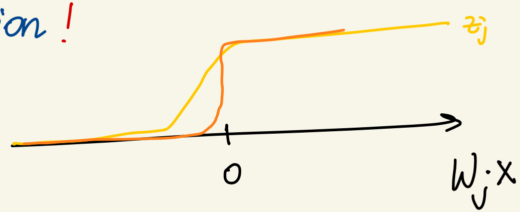
$$f = \sum \beta_j \overbrace{\varphi(W_j x)}^{z_j}$$

$$W_j^T \in \mathbb{R}^d \quad \text{or } \mathbb{R}^{d+1}$$

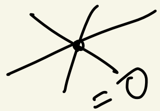
augmented x : $x = \begin{bmatrix} x_0 = 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$

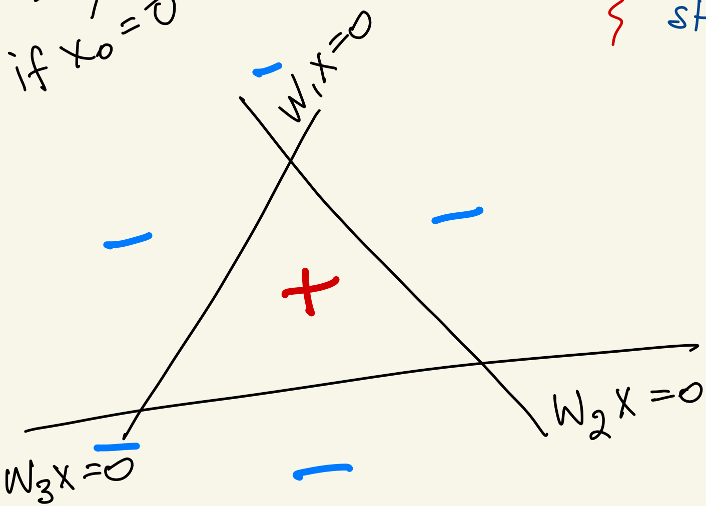


z_j is smoothed step function!



$\|W_j\| \nearrow \Rightarrow$ steeper change





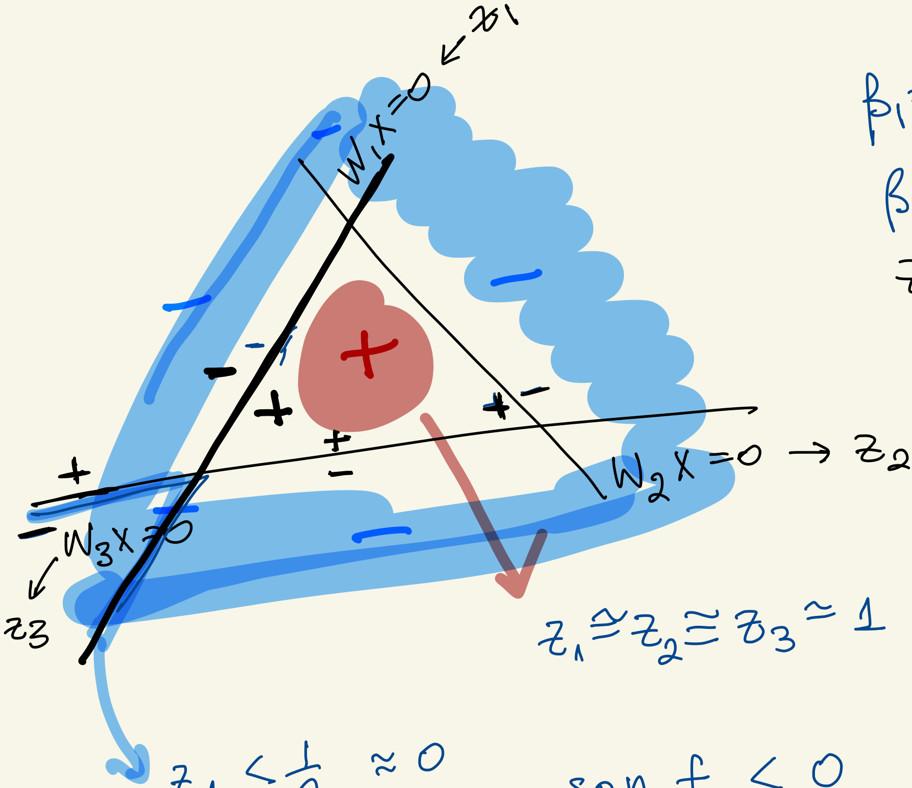
$$f = \beta_1 z_1 + \beta_2 z_2 + \beta_3 z_3 + \beta_4 \cdot z_4, \quad z_4 \equiv 1$$

classification : sign f

$x = [1 \; x_1 \; x_2]$

Exercise

$W_4 = ?$



$\beta_1 = \beta_2 = \beta_3 = 1$

$\beta_4 = -2.5 \leftarrow$ threshold

$z_4 \equiv 1$

$z_1 \simeq z_2 \simeq z_3 \simeq 1 \Rightarrow \text{sgn} f > 0$

$z_1 < \frac{1}{2} \approx 0$

$z_2 > \frac{1}{2} \approx 1$

$z_3 < \frac{1}{2} \approx 0$

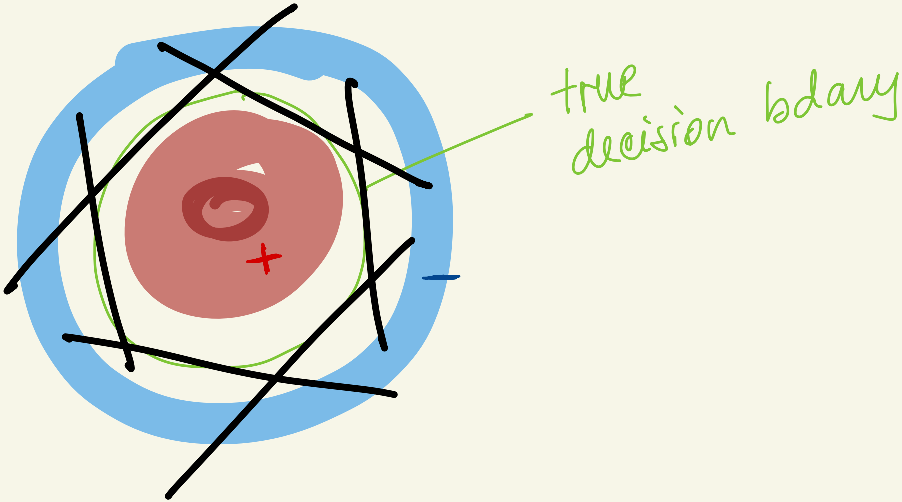
$\text{sgn} f < 0$

Increasing accuracy

- sharper transition $\Leftarrow$ increase $\|W_j\|$
- more complex decision boundary

Increasing accuracy

- sharper transition $\Leftarrow$ increase $\|W_j\|$
- more complex decision boundary $\Leftarrow$ increase m



$f = \beta^T z \quad \longrightarrow \quad \beta^T z + \beta_0$

OR $\tilde{z} = \begin{bmatrix} 1 \\ z_1 \\ \vdots \\ z_m \end{bmatrix}$, $\tilde{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}$ $\Big\} \rightarrow \tilde{\beta}^T \tilde{z}$

This page contains answers to some after class questions

What if no β_0? $\Rightarrow$ m larger !!

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

data $\quad x^{(1)}, \ldots x^{(n)} \in \mathbb{R}^d \quad d \times n$

$m, \varphi \longrightarrow \mathcal{F}_{m,\varphi}$ model class

$W, \beta \longrightarrow$ parameters or weights

Output layer options

- **linear** layer as in (2) $f = \sum_j \beta_j z_j$
- **logistic** layer: in classification $f(x) \in [0, 1]$ is interpreted as the probability of the $+$ class.

$$f(x) \;=\; \phi\left(\sum_{j=1}^{m} \beta_j z_j\right) \;=\; \phi\left(\sum_{j=1}^{m} \beta_j \phi(\sum_j w_{kj} x_k)\right) \tag{3}$$

- **softmax** layer in multiway classification

  The softmax function $\phi(z) : \mathbb{R}^m \to (0,1)^m$

$$\phi_k(z) \;=\; \frac{e^{z_k}}{\sum_{j=1}^{m} e^{z_j}} \tag{4}$$

- Properties
  - $\sum_{j=1}^{m} \phi_j(z) = 1$ for all z
  - for $z_k \gg z_j$, $j \neq k$ $\phi_k(z) \to 1$.
  - derivatives $\frac{\partial \phi_j}{\partial z_k} \;=\; \phi_k \delta_{jk} - \phi_j \phi_k$

Output layer

$z \in \mathbb{R}^m \longrightarrow y \in \mathcal{Y}$

- $y \in \mathbb{R}$ $\quad f(x) = \beta^T z$ $\quad$ regression $\quad$ Loss $= L_{LS}$

- $y \in \{\pm 1\}$ $\quad f(x) = \varphi(\beta^T z) - \frac{1}{2}$ $\quad$ binary classif $\Rightarrow \hat{y} = \operatorname{sgn} f = \operatorname{argmax}\{\beta^T z, -\beta^T z\}$

  Loss $= L_{logit} = -\ln P[y=+1|x]$

- $y \in \{1, 2, \dots r\}$ $\quad \hat{y} = \underset{k}{\operatorname{argmax}} \{\underbrace{\beta_1^T z, \beta_2^T z, \dots, \beta_r^T z}_{r \text{ outputs}}\}$

  multiway classification

parameters

W

$[\beta_1 \cdots \beta_r]$

$f(x) = \underbrace{\text{softmax}}_{\varphi}(\beta_1^T z, \dots \beta_r^T z) \in (0,1)^r$ (should be soft arg max !!)

Loss $= -\ln P_{y^{true}|x}$

$$\varphi_k(u) = \frac{e^{u_k}}{\sum_{k'=0}^{r} e^{u_{k'}}} \Rightarrow \varphi_k(u) \in (0,1),\ k = 1:r$$

$$\sum_k \varphi_k(u) = 1 \text{ for all } u \in \mathbb{R}^r$$

$k = 1:r$

Ex: $r = 2 \Rightarrow \varphi_1(u) =$ sigmoid (logistic)

$\varphi_2(1) = 1 - \varphi_1$

Interpret $\varphi_k(u) = Pr[u = k]$

$f_k(x) = Pr[y = k]$

model's confidence

- special cases of GLIM

Generalized Linear Models (GLM)

A GLM is a regression where the "noise" distribution is in the exponential fami ly.

- $y \in \mathbb{R}$, $y \sim P_\theta$ with

single variable y

$$P_\theta(y) \;=\; e^{\theta y - \ln \psi(\theta)} = \frac{1}{\psi(\theta)} e^{\theta y} \tag{5}$$

normalizing constant

- the parameter θ is a linear function of $x \in \mathbb{R}^d$

$P_{y|x} \Rightarrow \theta(x)$

$$\theta \;=\; \beta^T x \tag{6}$$

this W in NN

- We denote $\boxed{E_\theta[y] = \mu}$. The function $g(\mu) = \theta$ that relates the mean parameter to the natural parameter is called the **link function**.

The log-likelihood (w.r.t. β) is

FACT 1 obvious :

FACT 2, interesting:

$$l(\beta) \;=\; \ln P_\theta(y|x) \;=\; \theta y - \overset{\ln}{\psi}(\theta) \quad \text{where } \theta = \beta^T x \tag{7}$$

and the gradient w.r.t. β is therefore

$$\frac{\partial \ln P}{\partial \theta} = -\mu(\theta) + y$$

$$\nabla_\beta l \;=\; \boxed{\nabla_\theta l} \underbrace{\nabla_\beta(\beta^T x)} \;=\; (y - \mu) x \tag{8}$$

This simple expression for the gradient is the generalization of the gradient expression you obtained for the two layer neural network in the homework. [Exercise: This means that the sigmoid function is the *inverse link function* defined above. Find what is the link function that corresponds to the neural network.]

$$\frac{\partial}{\partial \beta}(\beta^T x) = \frac{\partial \theta}{\partial \beta} = x$$

Proof of FACT 2

$$\ln P(y) = \theta y - \ln \psi(\theta)$$

$$\psi(\theta) = \int_{-\infty}^{\infty} e^{\theta y}\, dy$$

$$\frac{\partial}{\partial \theta} \ln P_\theta y = y - \frac{\int y e^{\theta y} dy}{\psi} = y - \overbrace{\int y \underbrace{\frac{e^{\theta y}}{\psi(\theta)}}_{P_\theta(y)} dy}^{E_\theta[y]} = y - \mu(\theta)$$