Spring 2021

CSE 468.1

AI

Lecture 1

- Course Introduction
- Projects
- Questions

Course Objective and Outcome Form

Department of Electrical and Computer Engineering
School of Engineering and Physical Sciences
North South University, Bashundhara, Dhaka-1229, Bangladesh

1. **Course Number and Title:** **CSE 465 Pattern Recognition**
2. **Number of Credits:** **03**
3. **Type:** **Core**
4. **Prerequisites:** **CSE 373**
5. **Faculty Name:** **Dr. Nabeel Mohammed (NbM)**
6. **Room:** **SAC 917**
7. **Office Hours:** **TBA (Lets schedule as needed)**
8. **Email:** **nabeel.mohammed@northsouth.edu** + Google Classroom
9. **Contact Hours:** **Lectures – 3 Hours/week**
10. **Course Summary:**

    The course will cover topics such as supervised learning, unsupervised learning mainly in the context of neural networks. We will cover techniques, mainly popularised in the last few years, for neural network training and aim to give students an appreciation for the machine learning process involving this class of models.

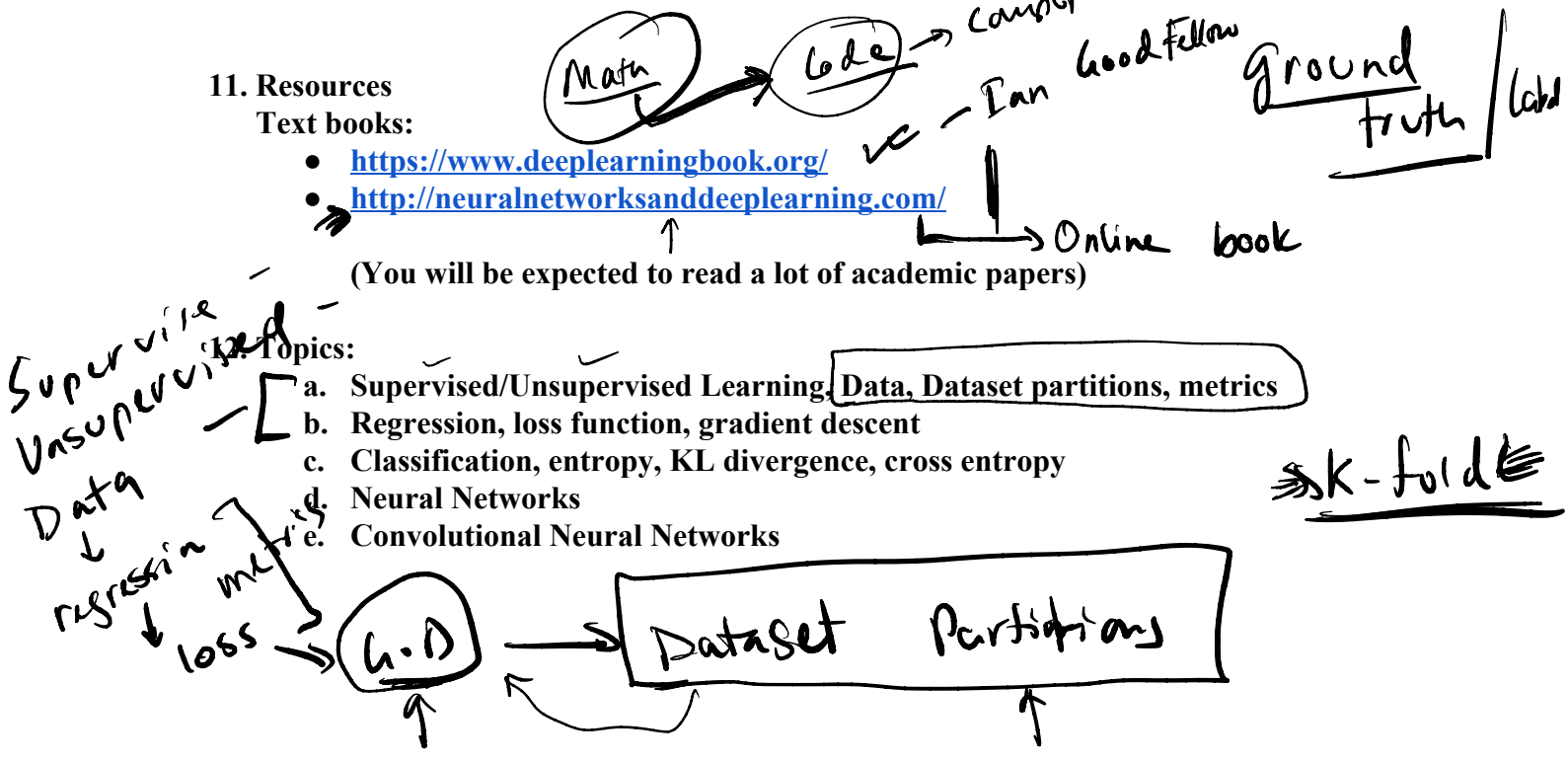
11. **Resources**

    Text books:

    - **https://www.deeplearningbook.org/**
    - **http://neuralnetworksanddeeplearning.com/**

    (You will be expected to read a lot of academic papers)

12. **Topics:**
    a. **Supervised/Unsupervised Learning, Data, Dataset partitions, metrics**
    b. **Regression, loss function, gradient descent**
    c. **Classification, entropy, KL divergence, cross entropy**
    d. **Neural Networks**
    e. **Convolutional Neural Networks**



f. **Recurrent Neural Networks**
g. **Autoencdoers**
h. **Generative Adversarial Networks**
i. **Siamese networks**
j. **Adversarial Images**
k. **Many other possible topics if time permits**

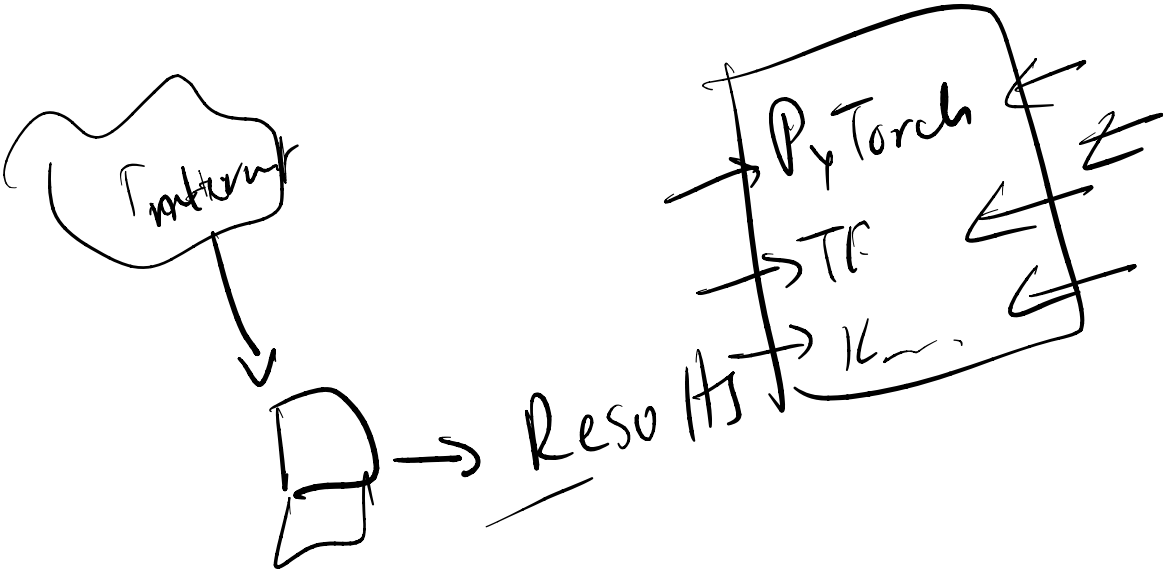
13. Weightage Distribution among Assessment Tools

Assessment Tools	Weightage (%)
Project Work	40
Literature Review of Project Topic (Due week 4)	10
Final Exam	25
Take home assessment (Mid)	25
	100

14. Grading policy: As per NSU grading policy available in http://www.northsouth.edu/academic/grading-policy.html

15. Course Policies:

a. Be good **and** follow university policies.



Project Ideas:

- Student should be able to frame a formal Machine Learning problem

- Student should be able to discern real world data & arrange data for model training, validation & testing.

- Choosing appropriate model, loss, & training, validation, testing strategis for model deployment

- Speaker Recognition

- Face Anti Spoofing

IEEE Access

Received February 26, 2020, accepted March 17, 2020, date of publication March 23, 2020, date of current version April 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982427

BANNER: A Cost-Sensitive Contextualized Model for Bangla Named Entity Recognition

IMRANUL ASHRAFI, MUNTASIR MOHAMMAD, ARANI SHAWKAT MAUREE, GALIB MD. AZRAF NIJHUM, REDWANUL KARIM, NABEEL MOHAMMED, AND SIFAT MOMEN

Department of Electrical and Computer Engineering, North South University, Dhaka 1229, Bangladesh

Corresponding author: Nabeel Mohammed (nabeel.mohammed@northsouth.edu)

This work was supported by the Special Grant of the Ministry of ICT, Bangladesh, under Grant 56.00.0000.028.20.004.19-248.

ABSTRACT Named Entity Recognition (NER) is a task in Natural Language Processing (NLP) that aims to classify words into a predetermined list of Named Entities (NE). Many architectures have produced good results on high resourced languages like English and Chinese. However, the NER task has not yet achieved much progress for Bangla, a low resource Language. In this paper, we perform the NER task on Bangla Language using Word2Vec and contextual Bidirectional Encoder Representations from Transformers (BERT) embeddings. We propose multiple BERT-based deep learning models that use the contextualized embedding from BERT as inputs and a simple statistical approach for class weight cost sensitive learning. The modified cost-sensitive loss function was used to address the class imbalance of the data. In our modified cost-sensitive loss function, we penalize the dominant classes by taking the ratio concerning the maximum sample in a class instead of the whole dataset. This penalty is made so that the learner learns slowly for the dominant class. In addition, we experiment by adding a Conditional Random Field (CRF) layer and incorporating Focal Loss to the training process. We found the best F1 Macro score to be 65.96%, F1 Micro score of 90.64%, and F1 Message Understanding Coreference (MUC) score of 72.04%, which were calculated at Named Entity level. Our experimental results demonstrate that one of the proposed models, which jointly optimizes for the CRF loss and class weighted cost-sensitive loss according to our proposed statistical approach, achieve an improvement of over 8% F1 MUC score on a recently introduced Bangla NER dataset when compared to previously published work.

INDEX TERMS Bangla NER, Bengali NER, BERT, CRF, NLP, focal loss, cost-sensitive learning, contextual embeddings.

I. INTRODUCTION

Named Entity Recognition (NER) is a task in Natural Language Processing (NLP) which seeks to classify words in an unstructured text into particular categories of Named Entities(NE), such as - person, organization, location etc. [18]. NER models can be used in multiple downstream tasks like identifying names of genes and gene products [43], recognition of chemical entities [50] and chatbots [40].

Bangla is a globally spoken language. Hence it is important that globally spoken languages like Bangla are enriched in linguistic knowledge and vocabulary. But current state of Bangla NLP has not achieved much progress in tasks like NER. This is because, unlike English, Bangla is more complex regarding both usability and vocabulary. Since Bangla is a rich language, many linguistic challenges occur while training models on it.

Figure 1 provides examples which illustrate the real challenges of Bangla language more precisely. Unlike English, there are minimal indicators for tags like capitalization in Bangla as stated by [17], [42]. For instance, under *Capitalization* in the figure, the words *'chottogram'* and *'kushtia'* refer to *Chittagong* and *Kushtia*, which are names of places. From the English words, it is clearly understood that *Chittagong* and *Kushtia* are capitalized and names (of locations). Hence, it is possible that they can be tagged as *Location* through NER models trained in English Language. On the other hand, observing the Bangla sentences, it is not clear as to which parts of speech do each word signify simply by looking at the words appearance without prior knowledge of Bangla.

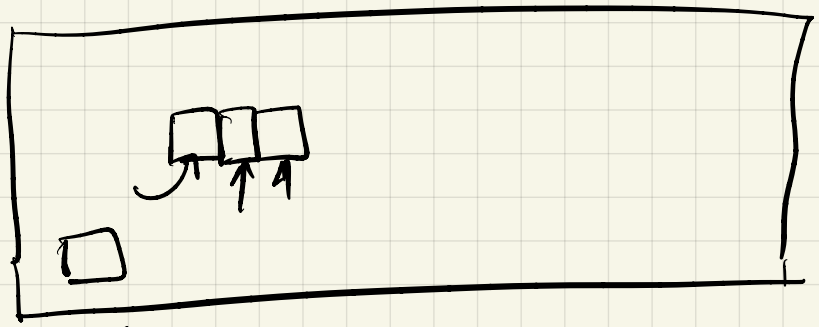
The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Tucci.

ML Engineer → X

↓

S. Engineer (Must be able to do ML)

- Project Ideas
  - Keyboard Layout optimization (generate)
  - Speaker Recognition ✓
  - Dhaka AI ✓
  - Bangla Named Entity Recognition ✓
  - Violence Detection ✓
  - Bangla H. Writing recognition
  - Machine Translation → Bangla



QWERTY

(metric)

NN

↳ Optimizations

....

Spell Corrections

↓

NN

→

Spell Correction

↓

Self Adapting

↓

Domain specific

→ Statistically learnt.

Lecture 2

$f(x) = x^2 + 2 \equiv ax^2 + bx + c$, $a = 1$, $b = 0$, $c = 2$

$f(1) = 1^2 + 2 = 3 \rightarrow (1, 3)$

Don't know the form of f

We don't know the parameters.

$\theta = (\quad)$

$f(x) = ax^2 + bx + c$

$\Downarrow$

$x^2 + 2$

$(a = 1, b = 0, c = 2)$

$\theta = (a, b, c)$

$\theta = (1, 0, 2)$

good form for f

x	1	2	3	4	5
Y	2	5	5	10	12

$\{(1,2), (2,5), (3,5), (4,10), (5,12)\}$

$Y = f(x) + n$ (*)

dependent $\hat{Y} = \hat{f}(x)$ Independent

Estimated function

$\hat{x} = g(x)$

$Y = g(x)$

image face

fingerprint

$\hat{g}$

x	1	2	3	4	5
y	2	5	5	10	12

$\hat{f}(x) = ax + b$

$\theta = (a, b)$

n

(V N)

Algorithm:

Purpose of this algorithm:

close

x	1	2	3	4	5	
y	2	5	5	10	12	•
$f(x) = x$, $\theta = (1,0)$	1	2	3	4	5	
	1	3	2	6	7	$\Sigma = 19$
$f(x) = x+1$, $\theta = (1,1)$	2	3	4	5	6	
	0	2	1	5	6	$\Sigma = 14$
$f(x) = 3x$, $\theta = (3,0)$	3	6	9	12	15	
	-1	-1	-4	-2	-3	$\Sigma = -11$
$f(x) = 2x$, $\theta = (2,0)$	2	4	6	8	10	
	0	1	-1	2	2	$\Sigma = 4$
$f(x) =$ magic(x), $\theta = (HP)$	2	5	5	10	12	
	0	0	0	0	0	$\Sigma = 0$

close

How close measurem

at

$1 + 9 + 4 + 36 + 49 = 99$

$(2-1)^2 + (5-2)^2 + (5-3)^2 + (10-4)^2 + (12-5)^2$

close

x	1	2	3	4	5		
y	2	5	5	10	12		
$\hat{f}(x) = x$, $\theta = (1,0)$	1	2	3	4	5	19 (A)	99 (S)
	1	3	2	6	7		
$\hat{f}(x) = x+1$, $\theta = (1,1)$	2	3	4	5	6	14 (A)	66 (S)
	0	2	1	5	6		
$\hat{f}(x) = 3x$, $\theta = (3,0)$	3	6	9	12	15	71 (A)	31 (S)
	1	1	4	2	3		
$\hat{f}(x) = 2x$, $\theta = (2,0)$	2	4	6	8	10	6 (A)	10 (S)
	0	1	1	2	2		
$\hat{f}(x) =$ magical, $\theta = (HP)$	2	5	5	10	12	0 (A)	0 (S)
	0	0	0	0	0		

absolute

min distance = 0 possible

total distance = function individual distances

individual distances ≥ 0

$\sum$ individual distances

$$\text{Total Distance} = \sum_{i=1}^{5} g(y_i, \hat{f}(x_i))$$

g is a distance function

$y, \hat{f}(x)$

$g(a,b) = |a-b|$ → Absolute error

$g(a,b) = (a-b)^2$ Squared error

An initial form of Gradient Descent

initialize a to some random value

while (not happy with error)
{
  calculate E.
  calculate $\frac{dE}{da}$

  Step = $\begin{cases} -1, & \frac{dE}{da} > 0 \\ 1, & \frac{dE}{da} < 0 \\ 0, & \frac{dE}{da} = 0 \end{cases}$

  $a = a + Step$

  Check happiness with new error.
}

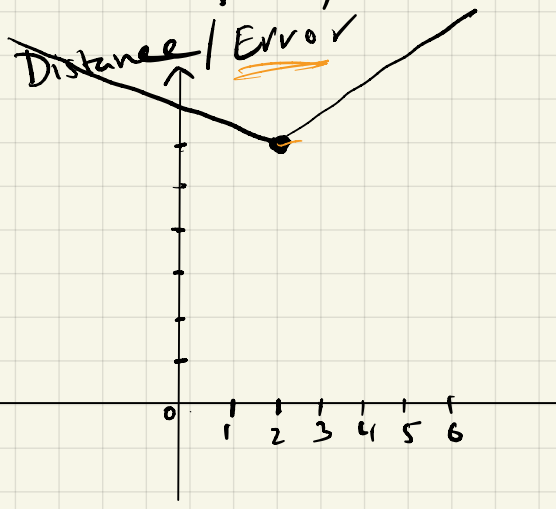
Data partitions

Distance function

$\hat{f}(x) = ax$

$\Theta = (a)$

$|y_i - \hat{f}(x_i)|$

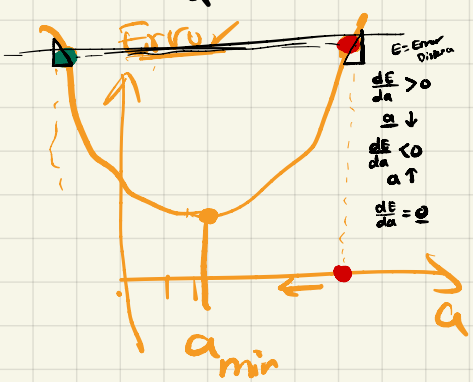


a :

Math

↓

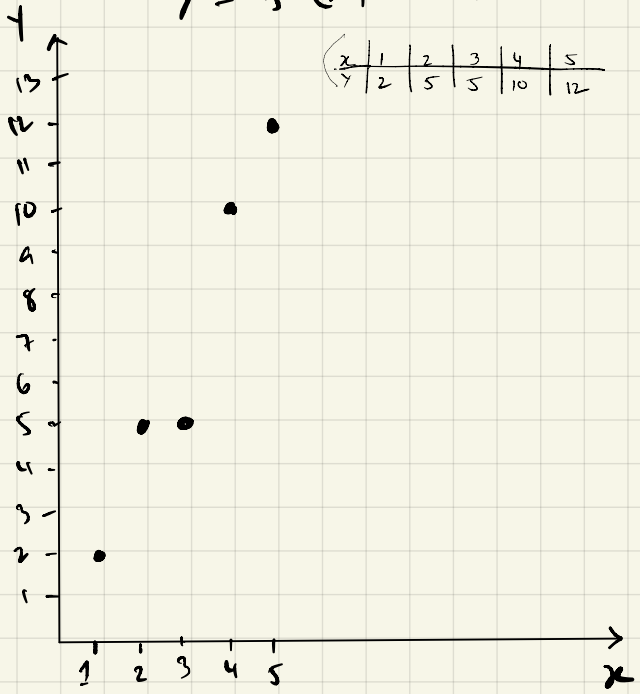
Computer ⇐ Reality



Lecture 3

x	1	2	3	4	5
y	2	5	5	10	12

$\hat{y} = \hat{f}(x) = ax$



initialize a randomly
while (not happy with error)
{
calculate error $E \rightarrow \sum_{i=1}^{5} (f(x_i) - y_i)^2$
calculate $\frac{dE}{da}$

step = $\begin{cases} -1, & \frac{dE}{da} > 0 \\ 1, & \frac{dE}{da} < 0 \\ 0, & \frac{dE}{da} = 0 \end{cases}$

$a = a \pm \text{step}$

check happiness with new error
}

Shopne Paben

$a = 5$

while (not happy error)

$E =$

Error $= (ax_1 - y_1)^2 +$
$(ax_2 - y_2)^2 +$
$(ax_3 - y_3)^2 +$
$(ax_4 - y_4)^2 +$
$(ax_5 - y_5)^2$

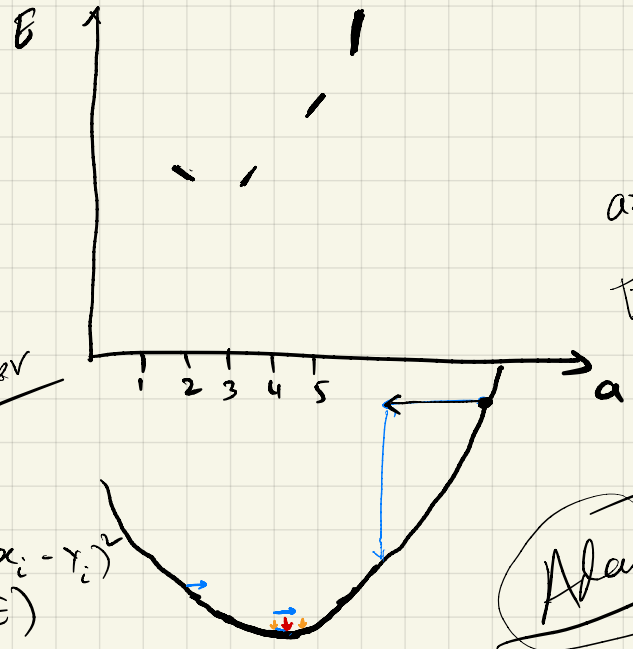
$= \sum_{i=1}^{5} (\hat{f}(x_i) - y_i)^2$

$= \sum_{i=1}^{5} (ax_i - y_i)^2$

a	E	dE/da
5	403	296
4	162	186
3	31	76
2	10	-34

$\frac{dE}{da} = 2(ax_1 - y_1) \cdot x_1 +$
$2(ax_2 - y_2) \cdot x_2 +$
$2(ax_3 - y_3) \cdot x_3 + 2(ax_4 - y_4) \cdot x_4 +$
$2(ax_5 - y_5) x_5 = \sum_{i=1}^{5} 2(ax_i - y_i) \cdot x_i$

a	E	dE/da
5	403	296
4	162	186
3	31	76
2	10	-34



learnable parameter

Initialize (a) randomly

Calculate $E = \frac{1}{5}\sum_{i=1}^{5}(ax_i - y_i)^2$

While (not happy with E)

{

calculate $\frac{dE}{da}$

Step = $\alpha * \frac{dE}{da}$

learning rate → Hyper parameter

$a = a - step$

ax

$f(x) = ax$

Constant

Adaptive

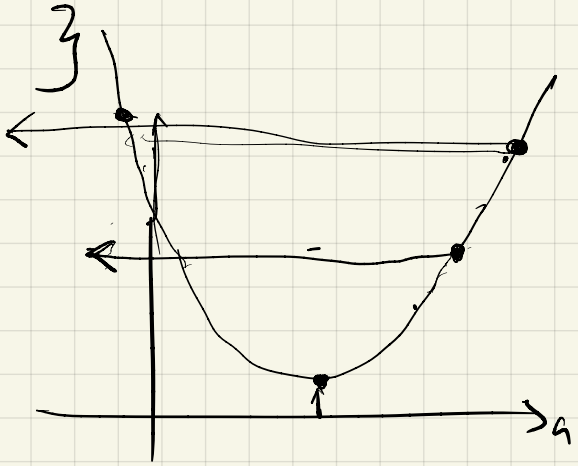
0.001

→ Experiment ati

$\alpha = 0.1$

calculate $E = \frac{1}{5}\sum_{i=1}^{5}(ax_i - y_i)^2$

x	1	2	3	4	5
y	2	5	5	10	12



a	E	dE/da	Step
5	80.6	59.2	5.92
-0.92	115.64	-71.04	-7.104
6.184			

0.1

x	$\hat{f}(x)$	$(y - \hat{f}(x))^2$
1	-0.92	$(2-(-0.92))^2 = 2.92^2 = 8.53$
2	-1.84	$(5-(-1.84))^2 = 6.84^2 = 46.78$
3	-2.76	$(5-(-2.76))^2 = 7.76^2 = 60.21$
4	-3.68	$(10-(-3.68))^2 = 13.68^2 = 187.$
5	-4.6	$(12-(-4.6))^2 = 275.56$

$\frac{dE}{da}$

$\frac{578.23}{5}$

Lecture 4

Generalizes

Trainings

$$

x	1	2	3	4	5	6	7	8
y	2	5	5	10	12	19	25·5	33
2·31x	2·31	4·62	6·93	9·24	11·55	13·86	16·17	18·48

$f(x) = ax$

validation

1-5

a	E
5	80·6
4	32·4
3	6·2
2	2
= 2·5	1·35
2·25	0·9875
2·31	0·9491

E
86·75
10·75
34·08
150
83
116
108

100's

0-1

10 × 10

MCD

test =

x	9	10	11
y	42	50	63
4x	36	40	44

Error = 165·67

Lecture 5

$Y = f(x) + \eta$, $\hat{Y} = \hat{f}(x)$, f is parameterized by θ

$Y, x, \hat{Y}$ → scalars

$Y, x, \hat{Y}$ (can be) is vectors.

$\hat{Y} = ax$, $x \in \mathbb{R}^3$, $Y \in \mathbb{R}^4$

$a \in \mathbb{R}^{4\times 3}$

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \hat{y}_4 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{34} \\ a_{41} & a_{42} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\hat{y}_1 = a_{11}x_1 + a_{12}x_2 + a_{13}x_3$$

$$\hat{y}_i = a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \quad \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \hat{y}_4 \end{bmatrix}$$

$$= \frac{1}{4}\left((\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + (\hat{y}_3 - y_3)^2 + (\hat{y}_4 - y_4)^2 \right)$$

For 1 training

Training data

Input X	$\vec{x}_1$	$\vec{x}_2$	$\vec{x}_3$	...	$\vec{x}_k$
Actual Y	$\vec{y}_1$	$\vec{y}_2$	$\vec{y}_3$	...	$\vec{y}_k$
Pred $\hat{Y}$	$\vec{\hat{y}}_1$	$\vec{\hat{y}}_2$	$\vec{\hat{y}}_3$	...	$\vec{\hat{y}}_k$

$(\hat{y} - y)^2$

$$\vec{x}_i = \begin{bmatrix} x_1^i \\ x_2^i \\ x_3^i \end{bmatrix} \in$$

$$\vec{y}_i = \begin{bmatrix} y_1^i \\ y_2^i \\ y_3^i \\ y_4^i \end{bmatrix} \quad \vec{\hat{y}} = \begin{bmatrix} \hat{y}_1^i \\ \hat{y}_2^i \\ \hat{y}_3^i \\ \hat{y}_4^i \end{bmatrix}$$

$$\frac{1}{k}\sum_{j=1}^{k} \frac{1}{4}\left[(\hat{y}_1^j - y_1^j)^2 + (\hat{y}_2^j - y_2^j)^2 + (\hat{y}_3^j - y_3^j)^2 + (\hat{y}_4^j - y_4^j)^2 \right]$$

$$= \frac{1}{Dk}\sum_{j=1}^{k}\sum_{d=1}^{D} (\hat{y}_d^j - y_d^j)^2 \quad \text{when } \hat{y}, \vec{y} \in \mathbb{R}^D$$

$$= \frac{1}{DK} \sum_{j=1}^{K} \sum_{d=1}^{D} \left(\hat{y}_d^j - y_d^j \right)^2 \quad \text{where } \hat{y}, y \in \mathbb{R}^D$$

$$= \frac{1}{DK} \sum_{j=1}^{K} m\left(\hat{f}(\vec{x}_j), y_j \right)$$

$$= \frac{1}{DK} \sum_{j=1}^{K} m\left(A\vec{x}_j, y_j \right)$$

$\vec{y} = A\vec{x}$

$\vec{z} = A(B\vec{x})$

.d

$$E = \frac{1}{DK} \sum_{j=1}^{K} \sum_{d=1}^{D} \left(a_{d1} x_1^j + a_{d2} x_2^j + a_{d3} x_3^j - y_d^j \right)^2$$

$$\frac{dE}{da_{pq}}$$

$\hat{y}^j =$

$A \in \mathbb{R}^{4 \times 3}$

d

initialize a Random A

Calculate E

while (not happy)

{

for each a_{pq} $(1 \le p \le 4, 1 \le q \le 3)$

{

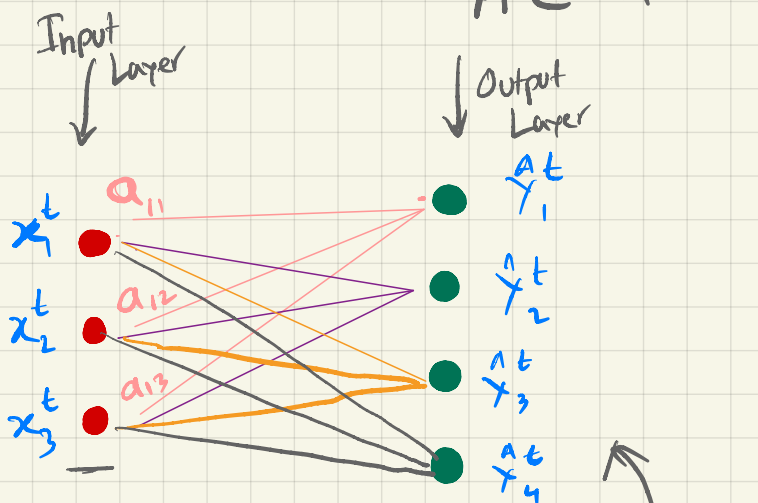
step = $\alpha \cdot \frac{dE}{da_{pq}}$

$a_{pq} = a_{pq} - \text{step}$

}

Calculate E

}



$\vec{x} \in \mathbb{R}^3$

$C\vec{x} \in \mathbb{R}^{100}$, $C \in \mathbb{R}^{100 \times 3}$

$\vec{\hat{y}} \in \mathbb{R}^4$, $B \in \mathbb{R}^{4 \times 100}$

4x100 x3

4x100

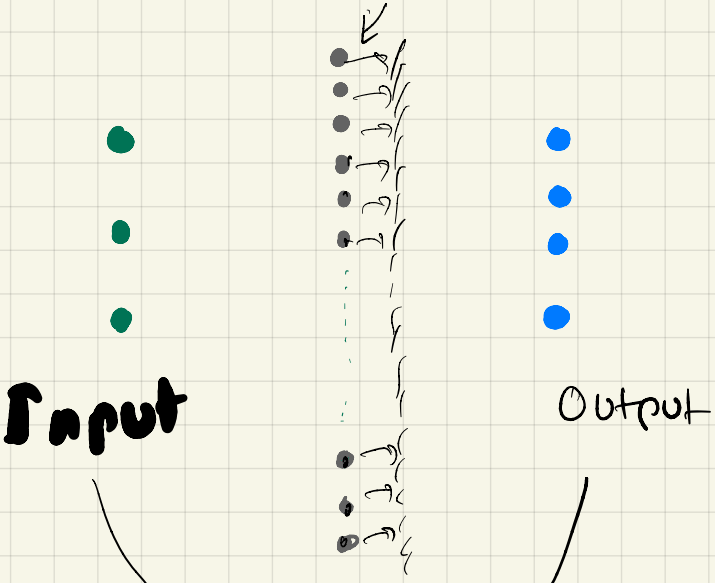
4x3

Neural Network

$\vec{\hat{y}} = A\vec{x}$

$\vec{\hat{y}} = B(C\vec{x})$

100



Linear
3 → 100
Linear
100 → 4
Bcx
$\downarrow\downarrow$
Tx

UAT

$\boxed{C\vec{x}} \rightarrow NL$

$B \cdot NL(C\vec{x})$

$T\vec{x}$

$$\hat{y} = B(C\vec{x})^2 \rightarrow \text{Elementwise Square}$$

or ↓ UAT

$B(C\vec{x})^3$ or $B \cdot \log(C\vec{x})$ or ⋮

E must be differentiable

Back Propagation

$$\hat{y} = A \cdot (B\vec{x})^2$$

$\hat{y}_{(2)} = ($ $)$

$x \in \mathbb{R}^3$

$\hat{x} \in \mathbb{R}^4$

$B \in \mathbb{R}^{5\times 3}$

$A \in \mathbb{R}^{4\times 5}$

Lecture 6

Back propagation

Context

Init θ randomly
~~Eta~~ Calculate E

θ is a ~~large~~ vector of all parameters

```
while (not happy)
{
    step = α dE/dθ   → partial derivatives
    θ = θ − step
    Calculate E
}
```

$x \in \mathbb{R}^3$ $\quad H_1 = \mathbb{R}^4$ $\quad \hat{x} = \mathbb{R}^2$ $\quad E = \frac{1}{D}\sum_i (y_i - \hat{x}_i)^2$

$H_1 = Sq(Ax)$ $\quad Sq$ - is elementwise square.

$\hat{x} = BH_1$

$A \in \mathbb{R}^{4\times 3}$

$B \in \mathbb{R}^{2\times 4}$

$$H_1 = Sq\left(\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right)$$

$$= \begin{bmatrix} (a_{11}\cdot x_1 + a_{12}\cdot x_2 + a_{13} x_3)^2 \\ (a_{21} x_1 + a_{22} x_2 + a_{23} x_3)^2 \\ (a_{31} x_1 + a_{32} x_2 + a_{33} x_3)^2 \\ (a_{41} x_1 + a_{42} x_2 + a_{43} x_3)^2 \end{bmatrix}$$

$$H_1 = \begin{bmatrix} (a_{11}x_1 + a_{12}x_2 + a_{13}x_3)^2 \\ (a_{21}x_1 + a_{22}x_2 + a_{23}x_3)^2 \\ (a_{31}x_1 + a_{32}x_2 + a_{33}x_3)^2 \\ (a_{41}x_1 + a_{42}x_2 + a_{43}x_3)^2 \end{bmatrix}$$

$$\hat{Y} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \end{bmatrix}$$

$$\begin{bmatrix} (a_{11}x_1 + a_{12}x_2 + a_{13}x_3)^2 \\ (a_{21}x_1 + a_{22}x_2 + a_{23}x_3)^2 \\ (a_{31}x_1 + a_{32}x_2 + a_{33}x_3)^2 \\ (a_{41}x_1 + a_{42}x_2 + a_{43}x_3)^2 \end{bmatrix} \rightarrow \begin{matrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{14} \end{matrix}$$

$$= \begin{bmatrix} b_{11}\cdot h_{11} + b_{12}\cdot h_{12} + b_{13}\cdot h_{13} + b_{14}\cdot h_{14} \\ b_{21}\cdot h_{11} + b_{22}\cdot h_{12} + b_{23}\cdot h_{13} + b_{24}\cdot h_{14} \end{bmatrix} \rightarrow \begin{matrix} \hat{Y}_1 \\ \hat{Y}_2 \end{matrix}$$

$$E = \frac{1}{2}\left((\hat{Y}_1 - Y_1)^2 + (\hat{Y}_2 - Y_2)^2\right)$$

$$12 + 8 = 20$$

$$\frac{dE}{da_{11}}$$

Big Annoying

$$E = k_{11}(a_{11})$$

$$E = k_{12}(a_{12})$$

$$E = \frac{1}{2}\left((\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2\right)$$

$$\frac{dE}{d\hat{y}_1} = 0 \quad \frac{1}{2} \cdot 2(\hat{y}_1 - y_1)$$

$$= (\hat{y}_1 - y_1)$$

x, y

$$\hat{y} = BSq(Ax)$$

$$\frac{dE}{d\hat{y}_1} = (\hat{y}_1 - y_1) \checkmark$$

$$\frac{dE}{d\hat{y}_2} = (\hat{y}_2 - y_1) \checkmark$$

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix}$$

$$\left[\frac{dE}{d\hat{y}_1} \quad \frac{dE}{d\hat{y}_2}\right]$$

$$\frac{dE}{db_{11}} = \frac{dE}{d\hat{y}_1} \cdot \frac{d\hat{y}_1}{db_{11}}$$

$$\frac{dE}{db_{12}} = \frac{dE}{d\hat{y}_1} \cdot \frac{d\hat{y}_1}{db_{12}}$$

$$\frac{dE}{db_{13}} = \frac{dE}{d\hat{y}_1} \cdot \frac{d\hat{y}_1}{db_{13}}$$

$$\frac{dE}{db_{14}} = \frac{dE}{d\hat{y}_1} \cdot \frac{d\hat{y}}{db_{14}}$$

RNN

$$\frac{dE}{dh_{11}} = \frac{dE}{d\hat{y}_1}\cdot\frac{d\hat{y}_1}{dh_{11}} + \frac{dE}{d\hat{y}_2}\cdot\frac{d\hat{y}_2}{dh_1}$$

$$= \_\_ \cdot b_{11} + \_\_ \cdot b_{21}$$

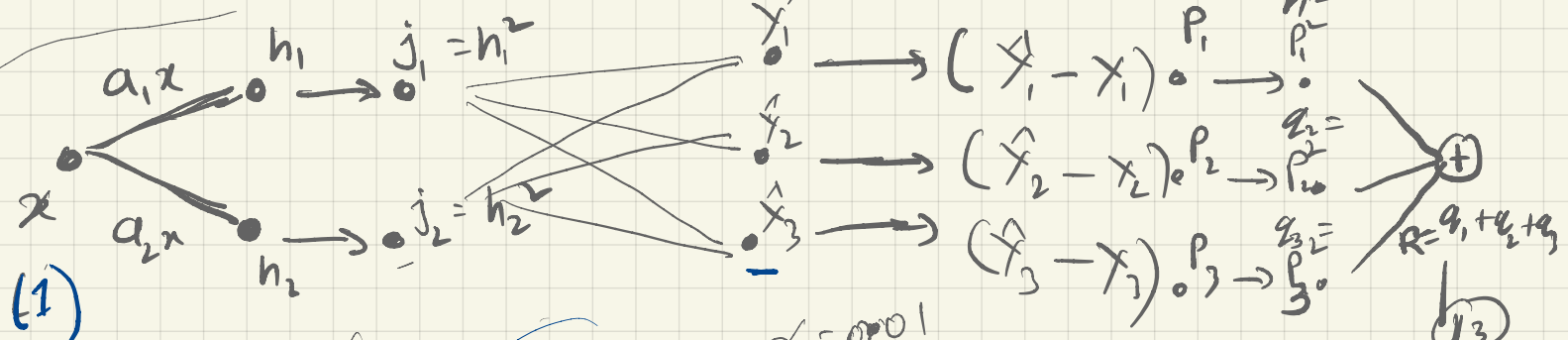
$$\frac{dE}{da_{11}} = \frac{dE}{dh_{11}} \cdot \frac{dh_{11}}{da_{11}}$$

$$h_{11} = (a_{11}\cdot x_1 + a_{12}\cdot x_2 + a_{13}\cdot x_3)^2$$

$$= 2\,(a_{11}x_1 + a_{12}\cdot x_2 + a_{13}x_3)\cdot x_1$$

$$\frac{dE}{da_1} = \left(\frac{dE}{dh_1}\right)\cdot\frac{dh_1}{da_1}$$

$$\frac{dE}{dh_1} = \frac{dE}{dj_1}\cdot\frac{dj_1}{dh_1}$$



(1)

$a_1 = 22$
$a_2 = 66$
$k_{11} = 6$
$k_{12} = 5$
$k_{21} = 0.9$
$k_{22} = 11$
$k_{31} = 5.3$
$k_{32} = -4$
$y_1 = 1000$
$y_2 = -1000$
$y_3 = 0$

$$H = \left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} x\right) = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$$

$$J = Sq(H)$$

$$\hat{y} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \\ k_{31} & k_{32} \end{bmatrix}\begin{bmatrix} j_1 \\ j_2 \end{bmatrix} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \end{bmatrix}$$

$\alpha = 0.001$

$x = 1$

- $h_1 = 22$, $h_2 = 66$
- $j_1 = (h_1)^2 = 484$, $j_2 = (h_2)^2 = 4356$
- $\hat{y}_1 = k_{11}\cdot j_1 + k_{12}\cdot j_2 = 24684$
- $\hat{y}_2 = k_{21}\cdot j_1 + k_{22}\cdot j_2 = 48351.6$
- $\hat{y}_3 = k_{31}\cdot j_1 + k_{32}\cdot j_2 = -148588$

$$\frac{dh_1}{da_{11}} = x = 1$$

$$\frac{dh_2}{da_2} = x = 1$$

$$\frac{dj_1}{dh_1} = 2h_1 = 44$$

$$\frac{dj_2}{dh_2} = 2h_2 = 132$$

$$\frac{dE}{dp_1} = \frac{dE}{d\hat{y}_1}\cdot\frac{d\hat{y}_1}{dp_1}$$

- $P_1 = \hat{y}_1 - y_1 = 23684$
- $P_2 = \hat{y}_2 - y_2 = 49351.6$
- $P_3 = \hat{y}_3 - y_3 = -148588$

$R = q_1 + q_2 + q_3 \approx 2.4\times10^9$

$E = R/3 = 8\times10^8$

$\frac{dE}{dR} = \frac{1}{3}$

$$\frac{dE}{dq_1} = \frac{dE}{dR}\cdot\frac{dR}{dq_1} = \frac{1}{3}\cdot 1 = \frac{dE}{dq_2} = \frac{dE}{dq_3}$$

- $q_1 = P_1^2 = 560.93\times10^6$
- $q_2 = P_2^2 = 2.4\times10^9$
- $q_3 = P_3^2 = 22078\times10^6$

$$\frac{dE}{dP_1} = \frac{dE}{dq_1}\cdot\frac{dq_1}{dP_1}$$

Lecture 7

- Supervised Learning
- Unsupervised Learning

Transfer Learning

- Classification
- Regression

M → Task 2

Task 1

$\hat{Y} = \hat{f}(x)$

Supervised

Unsupervised

(x_1, y_1)
⋮
(x_k, y_k)

$Y = f(x) + \Omega$

$\hat{y} = \hat{f}_\theta(x)$

$L = E(\hat{Y}, Y)$ ← Loss

$\frac{dE}{d\theta}$ → Gradient Descent

Image → Cat / Dog

Code

Human

(x_1, y_1) ---- (x_k, y_k)

Supervised Learning ← Ground Truth

Unsupervised

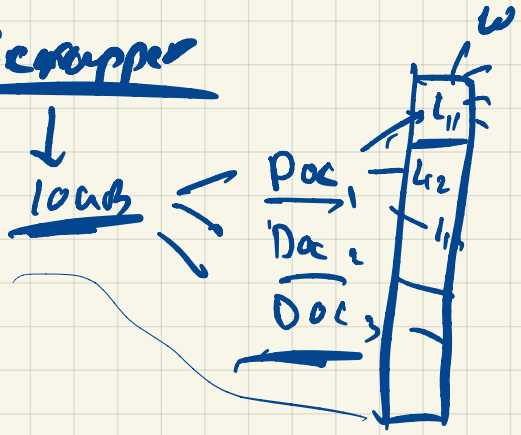
Image (B-W) → m → Color

(x_1, y_1)
⋮
(x_k, y_k)

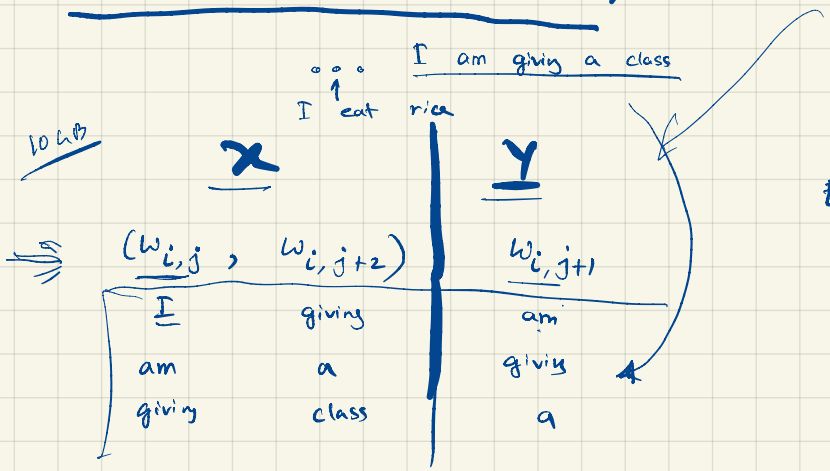
10,000

B-W

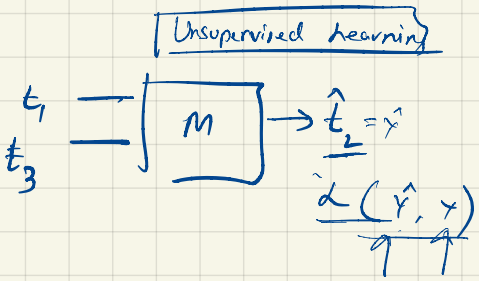
10 GB → Text Web Scrapper



$S_i = (w_{i,1} \quad \dots \quad w_{i,m})$



Unsupervised learning

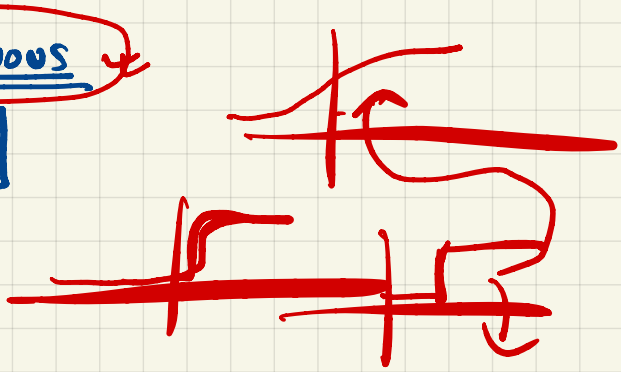


Classification vs Regression

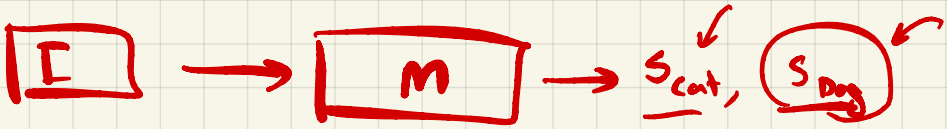
X → Discrete [:] 1

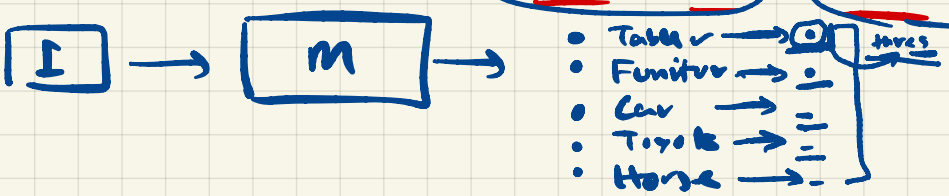
Continuous [:]

= BNL(Ax)

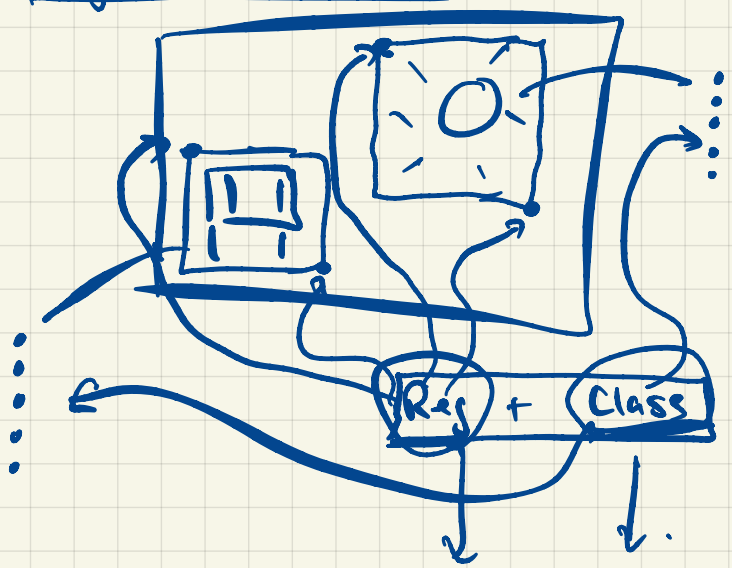


Interpretation

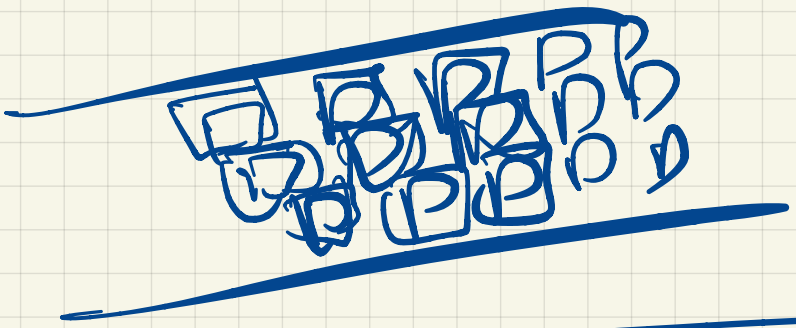




Object Detection



Yolo
↳
{FasterRENN
Efficient
Det



- Entropy (Information Theory)

P → I.T

M.S.E → classification → One Hot Encoding

P. Dist

⇒ {Entropy}

Lecture 8

PyTorch
Friday
2:30 pm

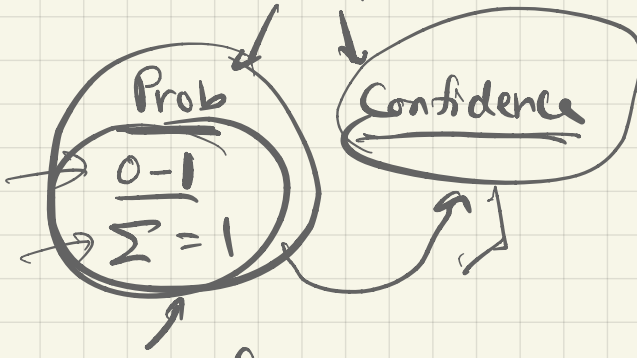
Classification

- 1 out of many classes

cat dog bird

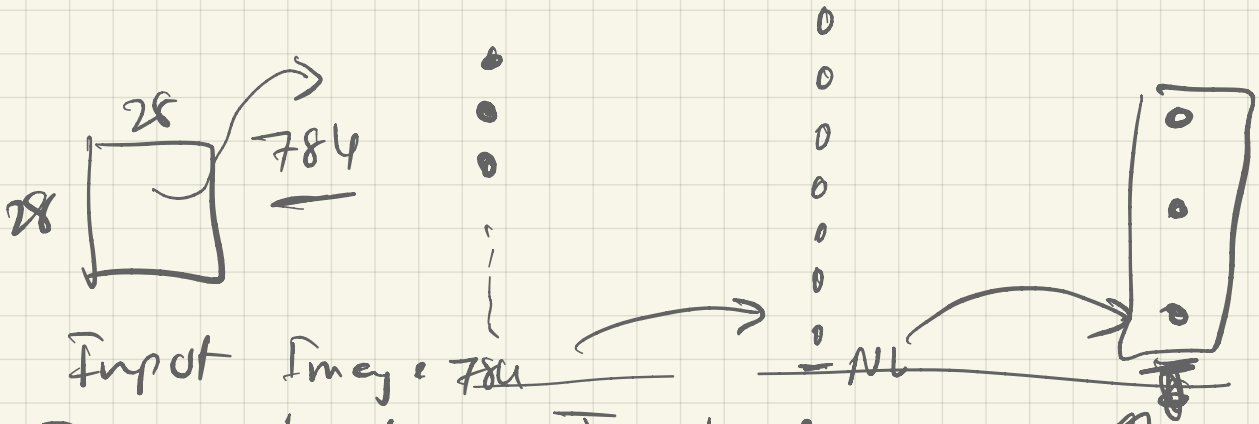
x → ↑ ↑ ↑

- Asking the model to predict scores for each class.



cat	dog	bird
321	500	786

- # of classes is fixed.



Input	Label	Transformed
x_1	Cat	$[1\ 0\ 0]^T$
x_2	dog	$[0\ 1\ 0]^T$
x_3	bird	$[0\ 0\ 1]^T$

One Hot representation

1 2 3

coordinate

$\hat{Y}_i = A \cdot NL(B\vec{x}_i)$ — Image

NL = Elem Wise Sq.

$\mathcal{L} = MSE(\hat{Y}_i, Y_i)$ — P.D

Y_i: [0 1 0] [1 0 0] [0 0 1] — one hot encoding

$\hat{Y}_i$: [3.5, 0.000001, 2 Billion]^T — P.D

→ 3000
→ 4000000
→ 3006
1

$\vec{x}_i \in \mathbb{R}^3$

[Each value 0-1]
$\Sigma = 1$

$\hat{Y}_i = S(A \cdot NL(B\vec{x}_i))$

$$S\left(\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}\right) = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

$0 \le P_j \le 1$

$\sum_{j=1}^{3} P_j = 1$

t_1 −200
t_2 16
t_3 50.2

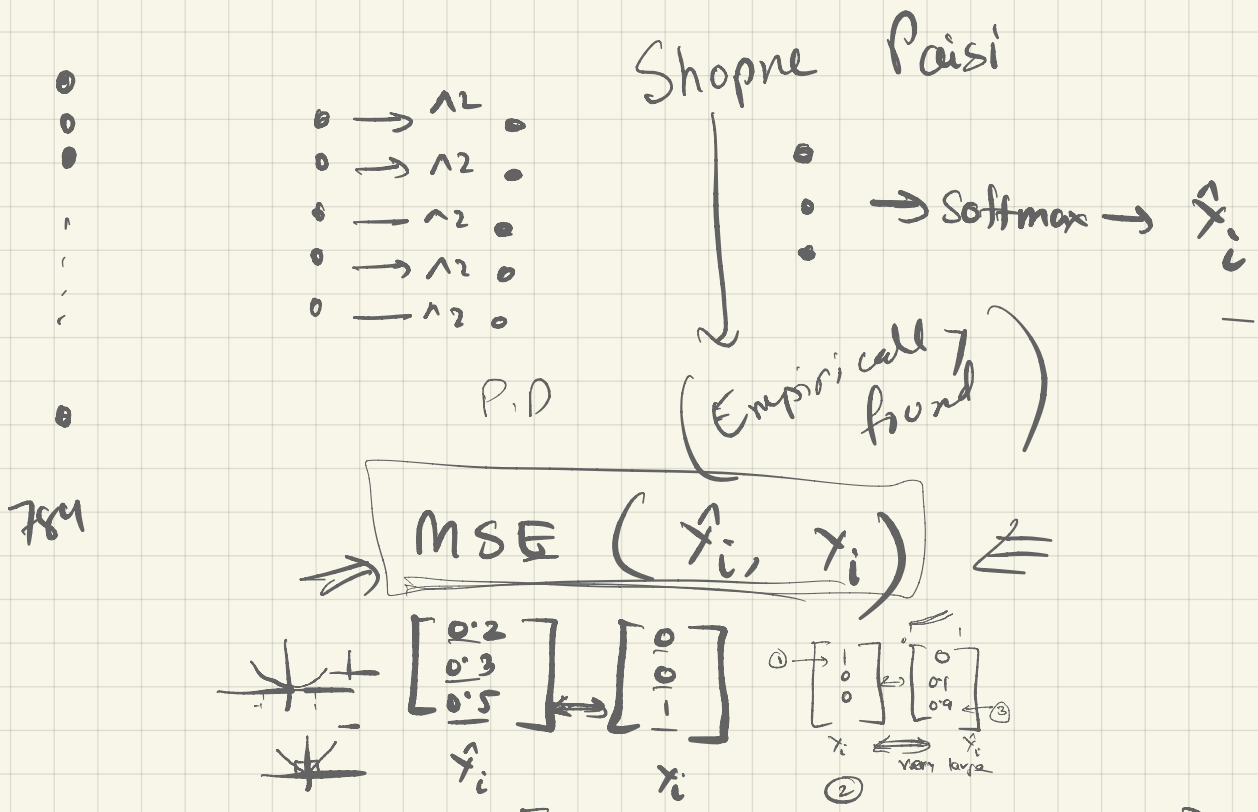
$$P_j = \frac{e^{t_j}}{\sum_{K=1}^{3} e^{t_K}}$$

(Softmax)

$t_1 \to \frac{e^{t_1}}{e^{t_1} + e^{t_2} + e^{t_3}}$

$t_2 \to \frac{e^{t_2}}{e^{t_1} + e^{t_2} + e^{t_3}}$

$t_3 \to \frac{e^{t_3}}{e^{t_1} + e^{t_2} + e^{t_3}}$



$$MSE(\hat{Y}_i, Y_i) = \frac{1}{3}\left[(0.5-1)^2 + (0.3-0)^2 + (0.2-0)^2\right]$$

$$= \frac{1}{3}\left[(-0.5)^2 + (0.3)^2 + (0.2)^2\right]$$

$$= \frac{1}{3}\left[(0.25) + (0.09) + (0.04)\right]$$

$$= \frac{1}{3}\left[0.38\right] = \underline{0.1267}$$

Measurement of distance between two P.D's..

⇓

K.L. Divergence a, b

{ entropy }

C.S. E

E2E

(Sfm1) (NLM)

C.S	Dr. Sifat Momen	Nabeel	⇒ Yusuf ⇐
Cake	0.2	0.3	0.333 ✓ K
Burger	0.5	0.3	0.333 ✓
Pizza	0.3	0.4	0.333 ✓

$L(\text{Cake}) = -\log 0.2$

$L(\text{Burger}) = -\log 0.5$

$L(\text{Pizza}) = -\log 0.3$

Average $L = \dfrac{-\log 0.2 + (-\log 0.5) + (-\log 0.3)}{3}$

(Sfm 1)

	Dr. Sifat Momen
Cake	0·2
Burger	0·5
Pizza	0·3

$L(\text{Cake}) = -\log 0{\cdot}2 = 2{\cdot}32$

$L(\text{Burger}) = -\log(0{\cdot}5) = 1$

$L(\text{Pizza}) = -\log(0{\cdot}3) = 1{\cdot}73$

$$\frac{1}{3}C + \frac{1}{3}B + \frac{1}{3}P$$

$$\frac{2{\cdot}32 + 1 + 1{\cdot}73}{3}$$

$$= 1{\cdot}68$$

C C B B B B B P P P

2·32 2·32 1 1 1 1 1 1·73 1·73 1·73

$$\frac{2C + 5B + 3P}{10} = 0{\cdot}2C + 0{\cdot}5B + 0{\cdot}3P$$

$$H(\text{Sifat}) = -0{\cdot}2 \log 0{\cdot}2 + (-0{\cdot}5 \log 0{\cdot}5) + (-0{\cdot}3 \log 0{\cdot}3)$$

$$= 1{\cdot}48$$

Lecture 9

$$-\sum_{i=1}^{3} s_i \log_2 s_i$$

$$-\sum_{i=1}^{3} n_i \log_2 n_i$$

	(Sfm1) Dr. Sifat Momen		(NbM) Nabeel		$H_s(N) = -\sum_{i=1}^{3} n_i \log_2 s_i$
Cake	0.2	S_1	0.3	n_1	
Burger	0.5	S_2	0.3	n_2	
Pizza	0.3	S_3	0.4	n_3	

S_C	$-\log_2 0.2 = 2.32$
S_B	$-\log_2 0.5 = 1$
S_P	$-\log 0.3 = 1.73$

N_C	$-\log_2 0.3 = 1.73$
N_B	$-\log_2 0.3 = 1.73$
N_P	$-\log_2 0.4 = 1.32$

$H(S) = 0.2 * 2.32 + 0.5 * 1 + 0.3 * 1.73$

$= 1.48$

$H(N) = 0.3 * 1.73 + 0.3 * 1.73 + 0.4 * 1.32$

$= 1.571$

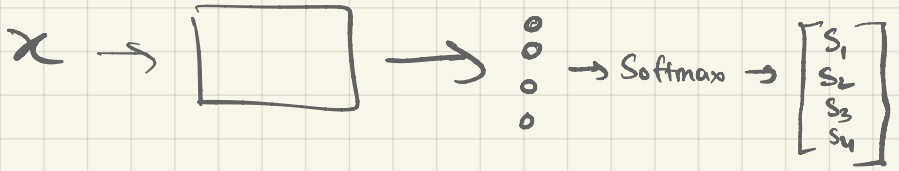
$H_S(N) = 0.3 * 2.32 + 0.3 * 1 + 0.4 * 1.73$

cross entropy $= 1.688$ Cost $= H_S(N) - H(N)$

Cost $KL_{S,N} = -\sum_{i=1}^{3} n_i \log_2 s_i - \left(-\sum_{i=1}^{3} n_i \log_2 n_i\right)$

$KL_{N,S} = -\sum_{i=1}^{3} s_i \log_2 n_i - \left(-\sum_{i=1}^{3} s_i \log_2 s_i\right)$

x	y
x_i	$[0\ 0\ 0\ 1]^T = y_i$



p G.T = $\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$ p_1 p_2 p_3 p_4 · Scores = $\begin{bmatrix}s_1\\s_2\\s_3\\s_4\end{bmatrix}$

Softmax

{ Sigmoid, relu } → Convolution networks

$$K.L_{S,G.T} = -\sum_{i=1}^{4} p_i \log_2 s_i - \left(-\sum_{i=1}^{4} p_i \log_2 p_i\right)$$

$$= -\log_2 s_4 + \log_2 1$$

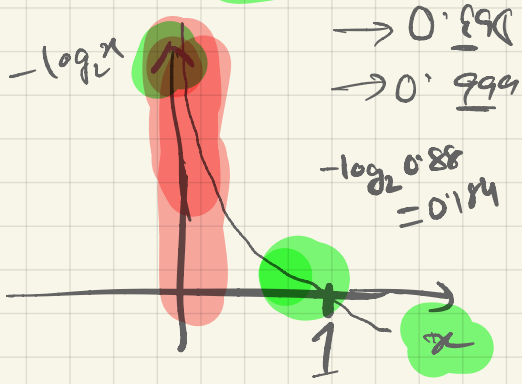
$$= -\log_2 s_4$$

Cross entropy loss
my → log likelihood loss
neg log loss, log loss

[ ] [ ]

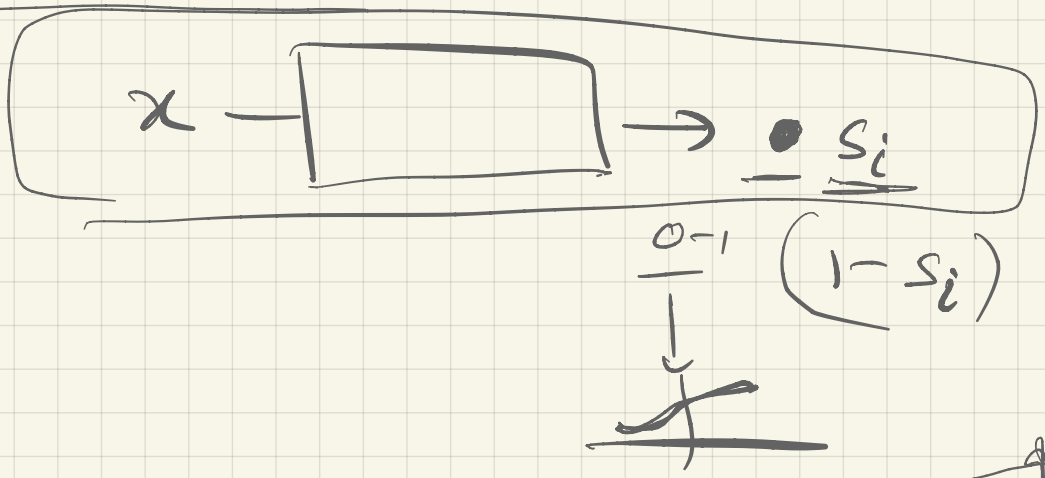
K.L

G.T	s_i	
1	0.001	$-\log_2 0.001 = 9.965$
	0.88	0.184

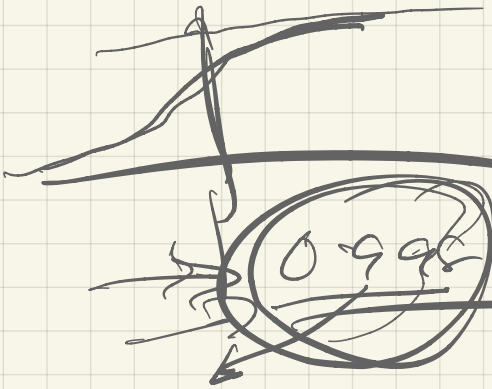


$$\alpha \cdot \frac{dE}{d\theta}$$

Binary Classification



x_i	G.T P_i (1/0)	Prediction s_i



$$-P_i \log_2 s_i - (1-P_i) \log_2 (1-s_i)$$

x	P_i	s_i	loss	
	1	0.99	Small (Zero)	0
	1	0.02	Big	α
	0	0.02	Small (Zero)	0
	0	0.9	Big	α

4 → $\frac{e^4}{e^4+e^3}$ = 3 → $\frac{e^3}{e^4+e^3}$ = x → [ ] → s_1, s_2

$\begin{bmatrix}0\\1\end{bmatrix}$ / $\begin{bmatrix}1\\0\end{bmatrix}$ P_1 P_2

$P_2 = (1-P_1)$

$S_2 = (1-S_1)$

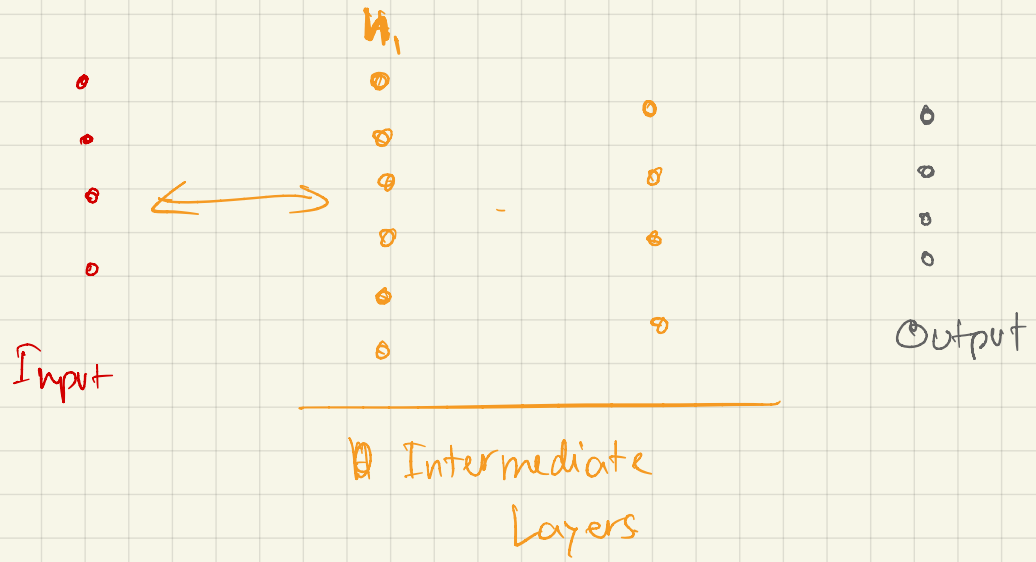
$$\alpha = -P_1 \log_2 s_1 - P_2 \log_2 s_2)$$

$$= -P_1 \log_2 s_1 - (1-P_1) \log_2 s_2$$

$$= -P_1 \log_2 s_1 - (1-P_1) \log_2 (1-s_1)$$

Binary cross entropy

Lecture 10



$$H_1 = A x$$

$$H_1^{NL} = \sigma_1(H_1)$$

$$H_2 = B H_1^{NL}$$

$$H_2^{NL} = \sigma_2(H_2)$$

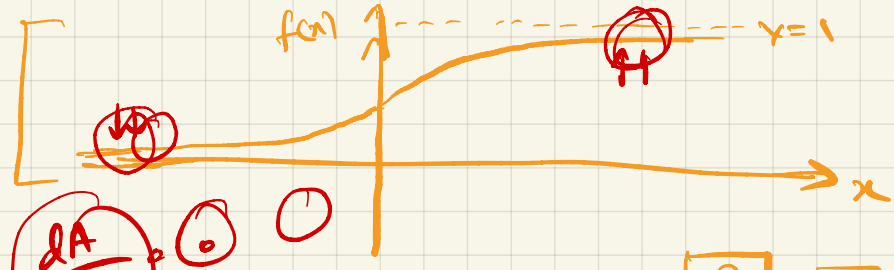
$$O_{score} = C H_2^{NL}$$

$$O_{prob} = \text{Softmax}(O_{score})$$

① Sigmoid

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

$e^{-(-50000)}$



$\frac{0-1}{\sim 0}$

$\frac{d\Lambda}{d\theta_1} = \frac{d\Lambda}{dA} \cdot \frac{dA}{d\theta_2}$

k → 0-1

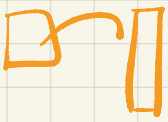
Score

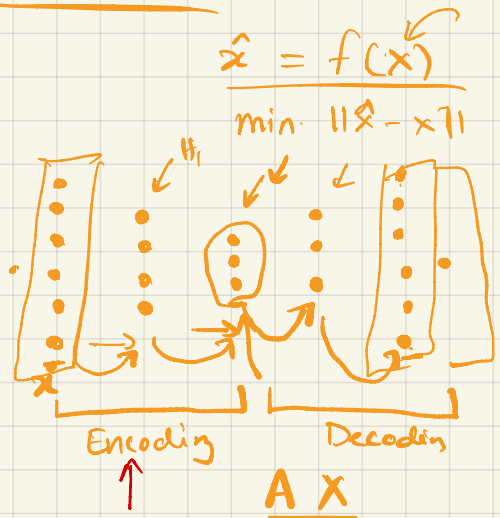
$\frac{d\Lambda_{final}}{d\theta_1}$

$\Lambda_{final} = \Lambda_1 + \Lambda_2 + \Lambda_3$

$\Lambda_{final} = \frac{1}{3}(\Lambda_1 + \Lambda_2 + \Lambda_3)$

Auto encoder



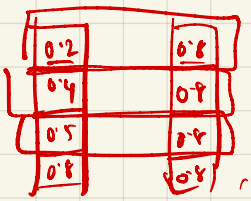


$$H_1 = A(x)$$
$$H_1^{NL} = \sigma_1(H_1)$$
$$H_2 = B H_1^{NL}$$
$$H_2^{NL} = \sigma_2(H_2)$$
$$H_3 = C H_2^{NL}$$
$$H_3^{NL} = \sigma_3(H_3)$$
$$H_4 = D H_3^{NL}$$
$$O = H_4^{NL} = \text{Sigmoid}(H_4)$$

Empirically

Color: 0-255 → 0-1

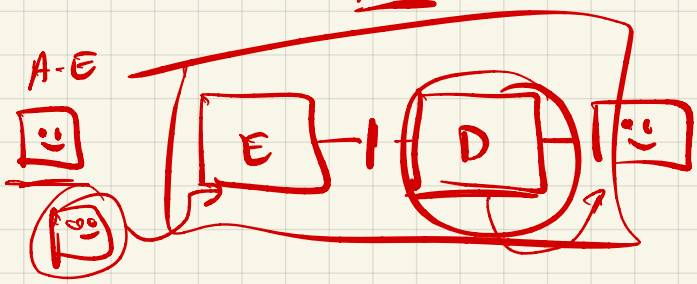
A X → 64 → H_1



mse

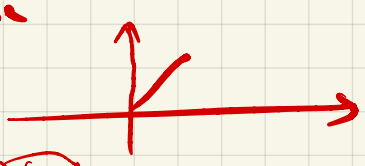
99% Data

768 → $\frac{1}{768}$



(2)

$$\text{Relu}(x) = \begin{cases} x, & x > 0 \\ 0, & \text{otherwise} \end{cases}$$

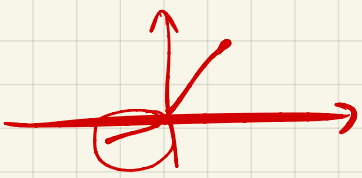


Relu (Ax)

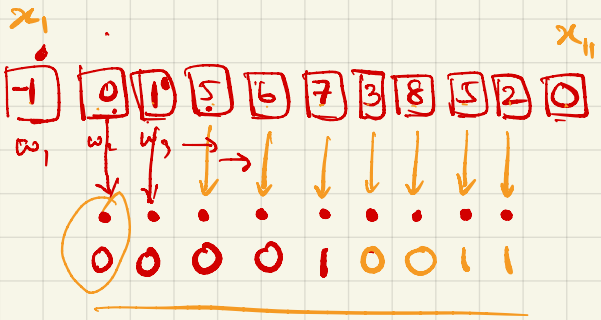
(3) Leaky Relu

$$\text{LRelu} = \begin{cases} x, & x > 0 \\ \alpha x, & x \leq 0 \end{cases}$$

α: 0·1



(4) $\tanh(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$



$[w_1\ w_2\ w_3]$

Sigmoid $(A\ X)$ ^{0}

arrows

11 column

R_1^T

R_2^T

$\begin{bmatrix} -1 \\ 0 \\ 1 \\ 5 \\ 6 \\ 7 \\ 3 \\ 8 \\ 5 \\ 2 \\ 0 \end{bmatrix}$ X

$-1-1 = -2$

99 parameters

$R_1 = \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

$R_1 \cdot X = -2$

Sigmoid $(-2) = 0.119 \approx 0$

$R_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

$R_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$