In GOD WE TRUST

Deep Learning Course

2021-2022 Fall Semester

HW01

...

Omid Sharafi
(400201518)

Instructor:
Dr.Emad Fatemizadeh

November 19, 2021

Sharif
University
of
Technology

Department of Electrical Engineering

Contents

1 Theoretical Problems

1.1 Vapnik-Chervonenkis

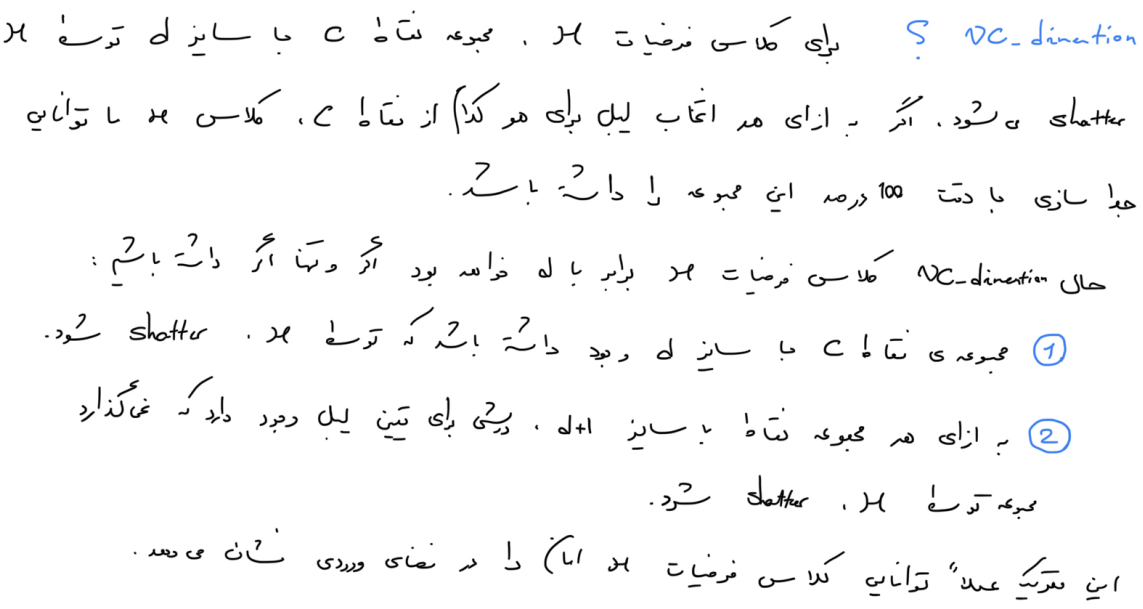
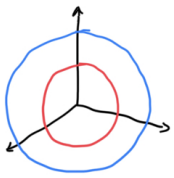


Figure 1: VC dimension

A: VC $\operatorname{sgn}(x^T x + \theta)$? (in 3D space)

answer: $VC = 1$



$$f(\vec{x}) = \begin{cases} \|x\|^2 + \theta < 0 \rightarrow - \\ \|x\|^2 + \theta > 0 \rightarrow + \end{cases}$$

It's obvious that for single node, VC can shatter it.
(just adjust θ)

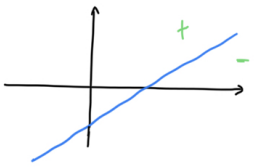
For 2 points x, y :

if $(\|x\| = \|y\|) \longrightarrow f(\vec{x}) = f(\vec{y})$: just Label $\vec{x}, \vec{y}$ different ✗

if $(\|x\| > \|y\|) \longrightarrow \|x\|^2 + \theta > \|y\|^2 + \theta$: $\begin{cases} \text{Label}(\vec{x}) = - \\ \text{Label}(\vec{y}) = + \end{cases}$ ✗

B: VC $\operatorname{sgn}(x_1\theta_1 + x_2\theta_2 + \theta_0)$?

answer: $VC = 3$



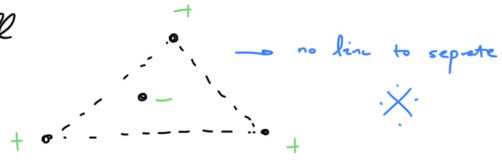
for 3 points :

→ at worst one label diffent from two others, so we can draw a line to separte that point ✓

for 4 points : if at least 3 points on a line

→ ✗

else, we have a convex hall



→ no line to separate ✗

Figure 2: Part A, B

C. VC superplane in N'D space ?

answer : $VC = N+1$

$f(x) = \text{sign}(w^T x + b)$, $w \in \mathbb{R}^N$, $b \in \mathbb{R}$

consider $N+1$ point : $\vec{P}_1 = \begin{bmatrix} 1 \\ \vdots \\ \vdots \end{bmatrix}$, $\vec{P}_2 = \begin{bmatrix} \vdots \\ 1 \\ \vdots \end{bmatrix}$, ... , $\vec{P}_n = \begin{bmatrix} \vdots \\ \vdots \\ 1 \end{bmatrix}$, $\vec{P}_{n+1} = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix}$

so for each point i $(1 \le i \le n)$ $f(x) = \text{sign}(w_i + b)$ → so for each labely we can assign perfect w_i and b ✓

for point $i = n+1$ $f(x) = \text{sign}(b)$

For upper bound : ① proof that set of $f(x) = \text{sign}(w^T x)$ can not shatter $N+1$

: assume that $S = \{x_1, \dots, x_{n+1}\}$ shattered by $f(x)$. so for each 2^{N+1} possible labeling, we have a proper w_k so that $f(w_k x)$ produce true labeling for all points in S.

$$H \triangleq \begin{bmatrix} w_1^T x_1 & w_2^T x_1 & \cdots & w_{2^{N+1}}^T x_1 \\ \vdots & \vdots & & \vdots \\ w_1^T x_{n+1} & w_2^T x_{n+1} & & w_{2^{N+1}}^T x_{d+1} \end{bmatrix}$$

each of the columns make our labeling

linear independant of rows

$\Rightarrow$ no v that $v^T H = 0$

output labeling$_k$ = kth entry $v^T H = v^T X w_k$

$1 < k \le 2^{N+1}$ ⟹ there is a k that $\text{sign}(X w_k) = \text{sign}(v)$ ⇒ $v^T X w_k > 0$

⇒ rank (H) = number of rows = $N+1$

$H = XW$ ⇒ rank $(H) \le \min\{\text{rank}(X), \text{rank}(W)\} \le N$ ✗ ($\le N$ we are inside $\mathbb{R}^N$ space)

② now we have $x = \begin{bmatrix} x' \\ 1 \end{bmatrix}$, $w = \begin{bmatrix} w' \\ b \end{bmatrix}$ and maximum VC_dimention of a $f(x') = \text{Sign}(w'^T x') \le N$, so VC_dimention $f(x) \le N+1$ /

Figure 3: Part C

1.2 MSE

Problem 2. MSE as loss function in regression problem $\xrightarrow{\text{same as}}$ Assuming normal distribution of data in ML problem

- MSE as loss function : $\boxed{w_{LS} = \underset{w}{\operatorname{argmin}} \sum (y_i - w^T x_i)^2}$
- normal distribution assumption : $y_i = w^T x_i + \varepsilon_i \;,\quad \varepsilon_i : \mathcal{N}(\cdot, \sigma^2)$

$\xrightarrow{\text{solve ML}}$ $w_{ML} = \underset{w}{\operatorname{argmax}} \prod f(y_i \mid w, x_i, \sigma) =$

$\underset{w}{\operatorname{argmax}} \prod \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(y_i - w^T x_i)^2}{2\sigma^2}} \overset{\ln}{=\!=} \underset{w}{\operatorname{argmax}} \sum \left(\underbrace{-\ln(\sqrt{2\pi}\,\sigma)}_{\text{cte}} - \frac{(y_i - w^T x_i)^2}{\underbrace{2\sigma^2}_{\text{cte}}} \right)$

$= \underset{w}{\operatorname{argmax}} \sum -(y_i - w^T x_i)^2 = \underset{w}{\operatorname{argmin}} \sum (y_i - w^T x_i)^2$ ✓

1.3 ML

Problem 3. calculate normal distibution parameters of $\{x_i\}_{i=1}^{n}$ with MSE

$$(\mu,\sigma)_{ans} = \underset{\mu,\sigma^2}{\arg\max} \prod f(x_i|\mu,\sigma^2) = \underset{\mu,\sigma^2}{\arg\max} \prod \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

$$\overset{\ln}{=} \underset{\mu,\sigma^2}{\arg\max} \sum \left(-\ln(\sqrt{2\pi}\sigma) - \frac{(x_i-\mu)^2}{2\sigma^2}\right)$$

$$\frac{\partial}{\partial\mu} = 0 \Rightarrow \frac{1}{2\sigma^2}\sum 2(x_i-\mu) = 0 \Rightarrow n\mu = \sum x_i \Rightarrow \boxed{\mu = \frac{1}{n}\sum x_i}$$

$$\frac{\partial}{\partial\sigma^2} = 0 \Rightarrow \sum \frac{-1}{2\sigma}\frac{1}{\sigma} + \frac{1}{2\sigma}\frac{(x_i-\mu)^2}{\sigma^3} = 0$$

$$\Rightarrow \frac{n}{2\sigma^2} = \frac{1}{2\sigma^4}\sum(x_i-\mu)^2 \Rightarrow \boxed{\sigma^2 = \frac{1}{n}\sum(x_i-\mu)^2}$$

1.4 Regression

Problem 4.

A) Regression : $w_{ans} = \underset{w}{\operatorname{argmin}} \; (Y - W.X)^T (Y - W.X)$

$= \underset{w}{\operatorname{argmin}} \; Y^T Y - Y^T (W.X) - (W^T X^T) Y + (W^T X^T)(W.X)$

$\frac{\partial}{\partial w} = -Y^T X - X^T Y + X^T (X.W) + (W^T X^T) X = 0 \xrightarrow{\text{scaler}} X^T (X.W) = X^T Y \Rightarrow \boxed{w = (X^T X)^{-1} X^T Y}$

B) if $X^T X$ has no inverse

$w_{asn} = \underset{w}{\operatorname{argmin}} \; \|Y - W.X\|_2^2 + \lambda \|W\|_2^2$ (Regularization L_2)

$\frac{\partial}{\partial w} = -2 X^T Y + 2 X^T X W + 2 \lambda W = 0 \Rightarrow \boxed{W = (X^T X + \lambda I)^{-1} X^T Y}$

2 Finger spelling

2.1 Visualize data set

In this part, we separated train, validation data. Made the output in one hot format. Shuffled the data and finally normalized the data of each part of the data set.

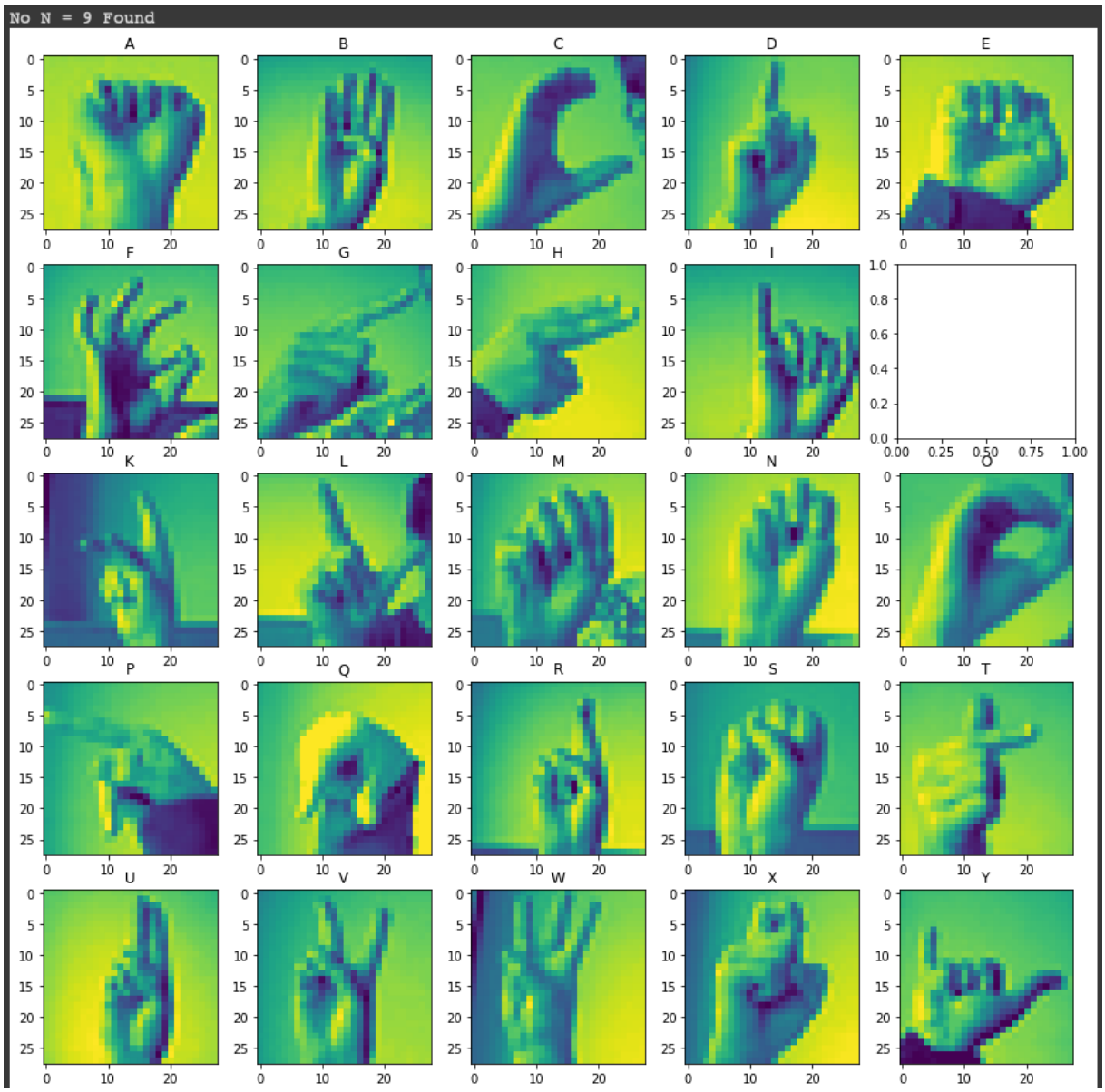


Figure 4: Finger Spelling Data set

2.2 Train model and draw loss and weights histogram

2.2.1 SGD optimizer result

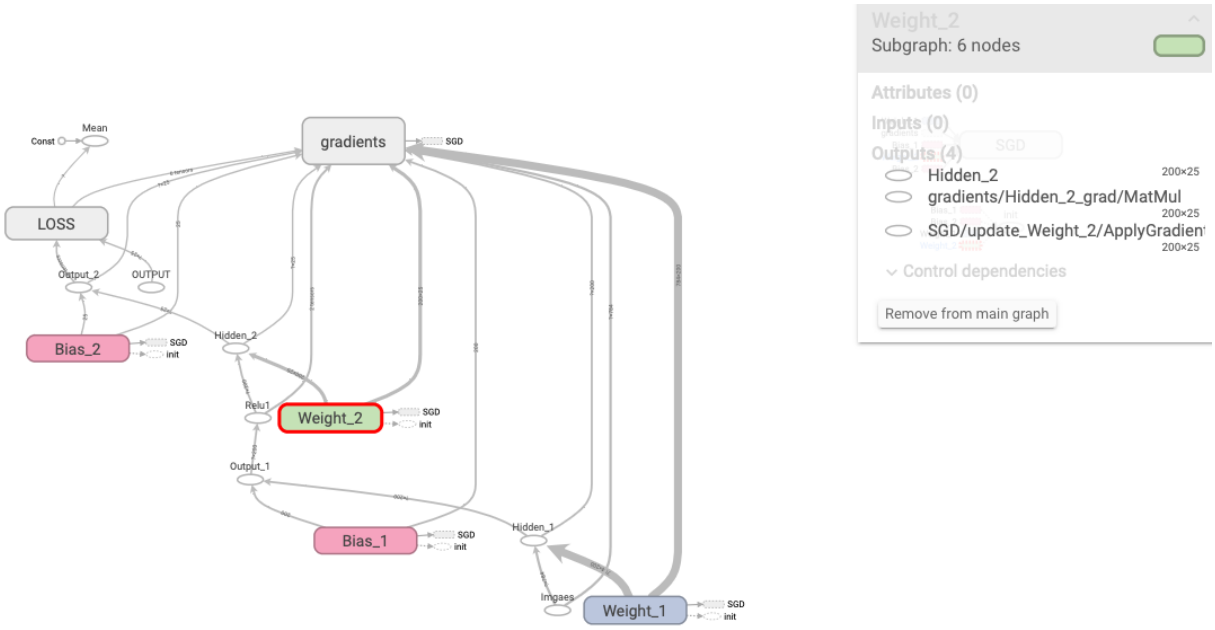


Figure 5: SGD network design(one 200 node hidden layer)

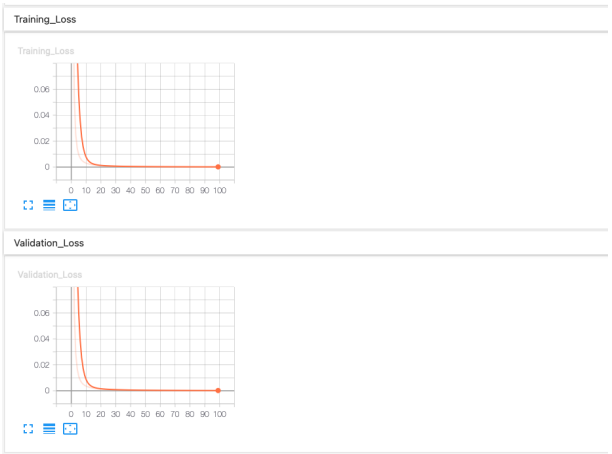


Figure 6: Loss

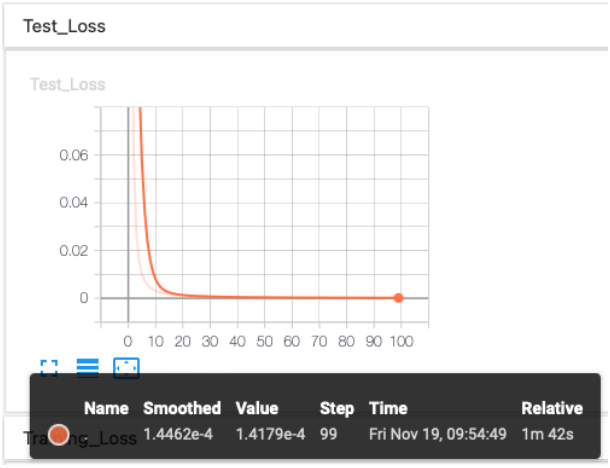


Figure 7: Loss

Figure 8: Accuracy

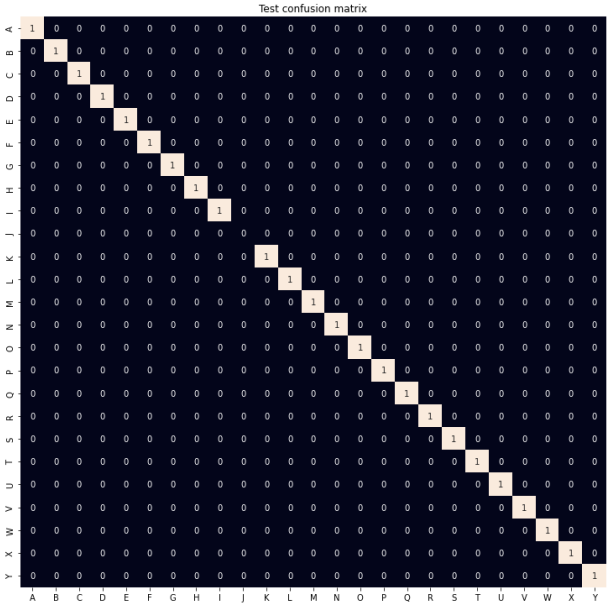


Figure 9: Confusion matrix

As you can see, the network learn data perfect and it is not over-fitting because the accuracy on the test is also perfect. Also to check if there is a any problem in training method or not I just reduced the epochs number and increased batch size and you can see the result at results folder. More general and deeper network is available at the end of the report and saved in the best folder.

And about the weighs distribution, you can see change its distribution at first epochs.

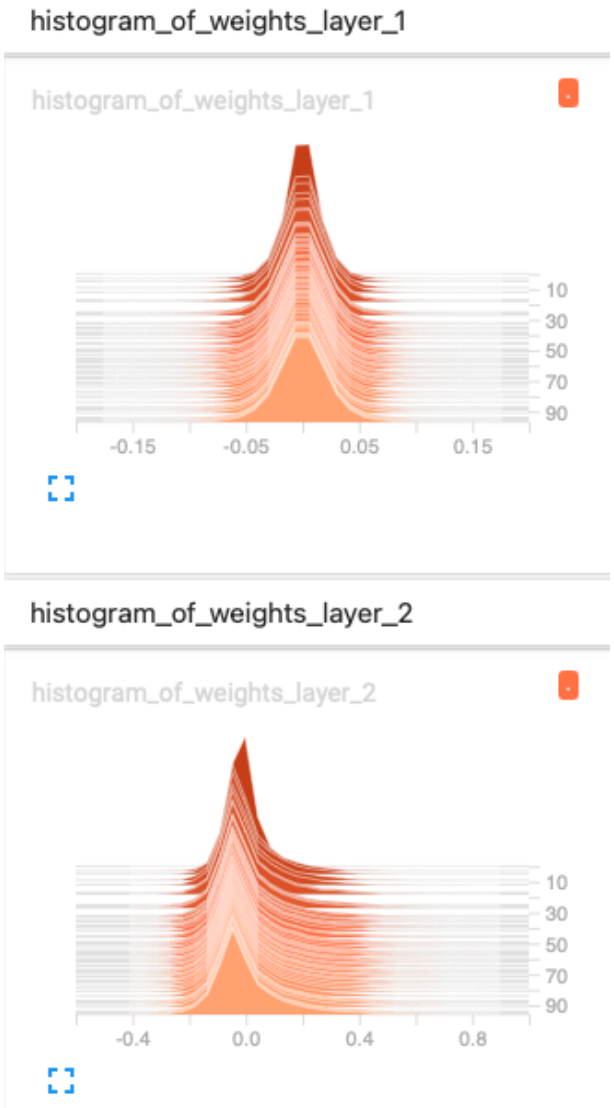


Figure 10: Weighs distribution

2.2.2 ADAM optimizer result

As you can see, both ADAM and SGD results are good. Although ADAM is more faster and better but it has lots of oscillation in its accuracy and convergence. For more approaches for example we can use NADAM, AdaGrad and other training approaches.

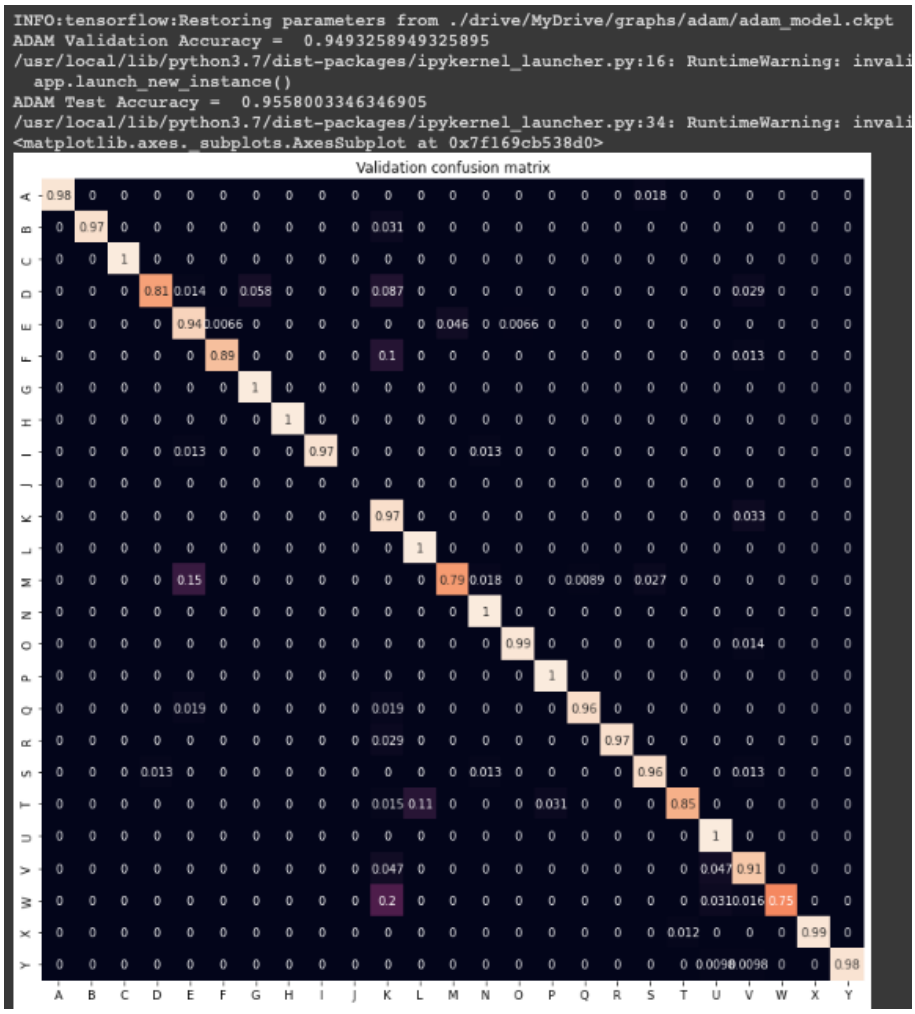


Figure 11: Confusion matrix (ADAM)

And about the weighs distribution, you can see change in distribution and increment of variance and separation during epochs in ADAM method.

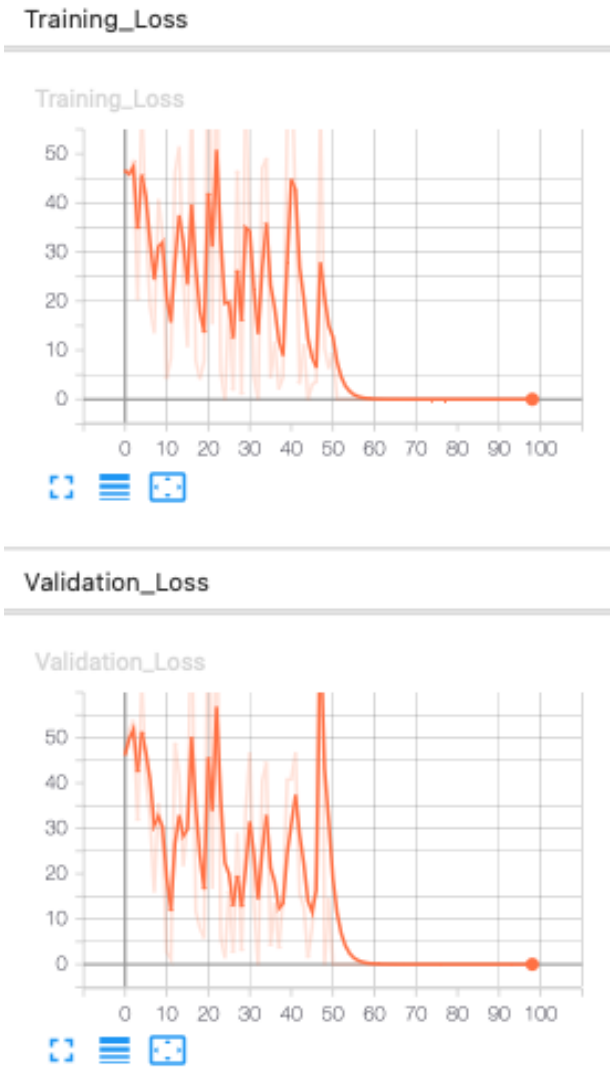


Figure 12: Loss (ADAM)

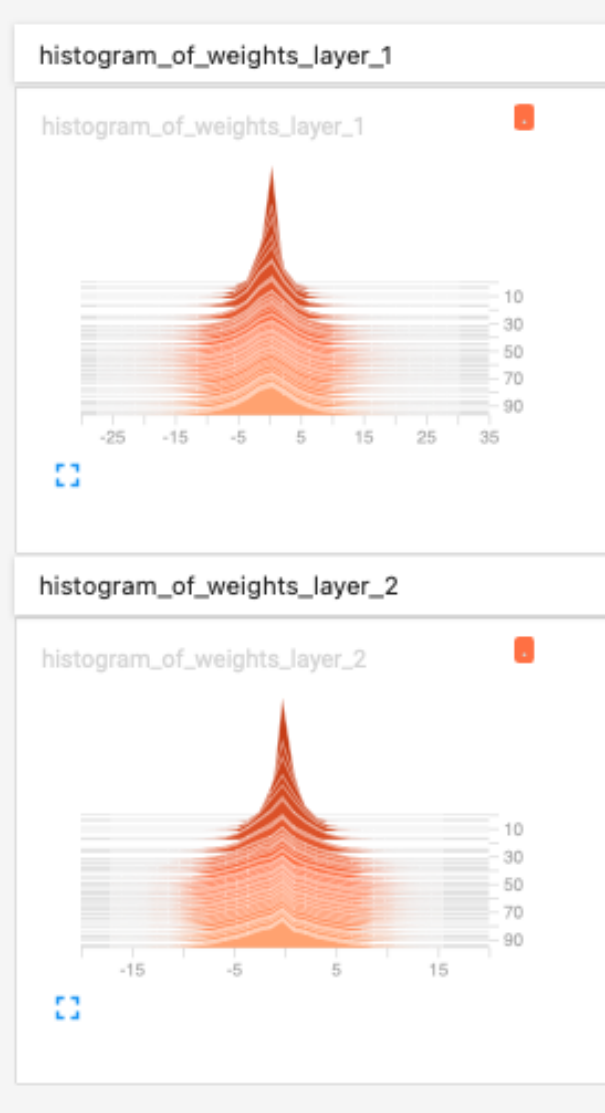


Figure 13: Weighs distribution (ADAM)

2.2.3 Dropout

From this part, to see the change result first we try to reduce our normal SGD network accuracy by increasing batch-size and run network just 5 epochs. Then I tried different combination of dropout rate and epochs numbers and its really fun that when we have:

$$\frac{epochsnumber_{new}}{epochsnumbers_{base}} = \frac{1}{dropoutrate}$$

then we can see some improvement in result. I mean that it seams that number of training on data is constant but one approach without dropout and the other one with dropout. Also the drop out rate should not be high.

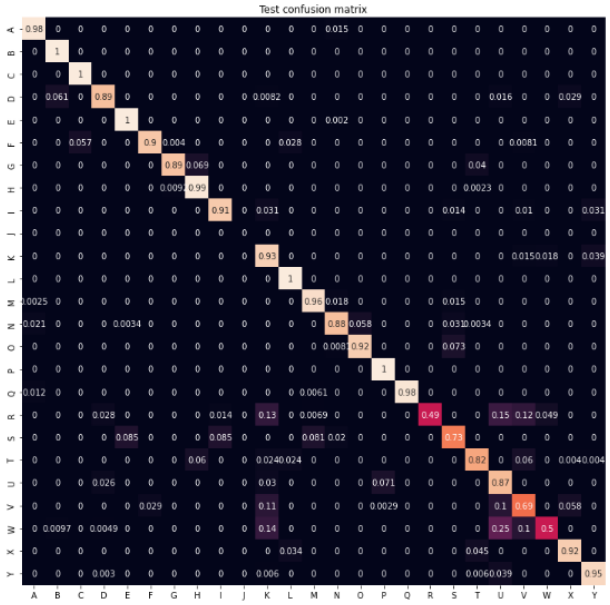


Figure 14: SGD without Dropout (Acc = 90.4)

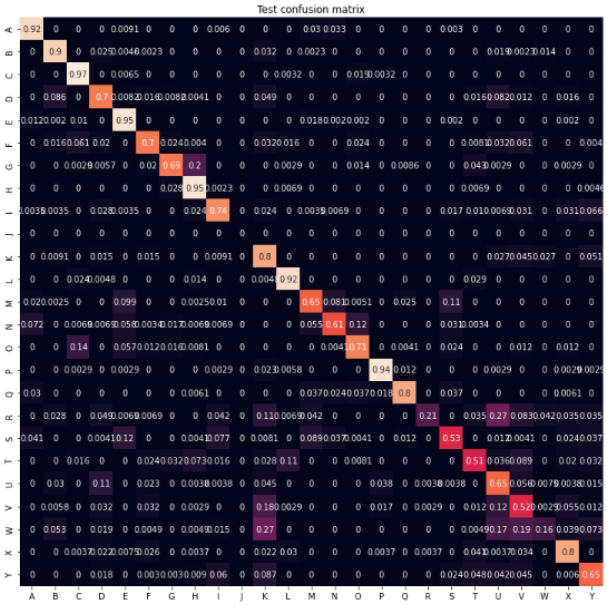


Figure 15: Dropout (5 ephoces with rate = 0.5) (Acc = 73)

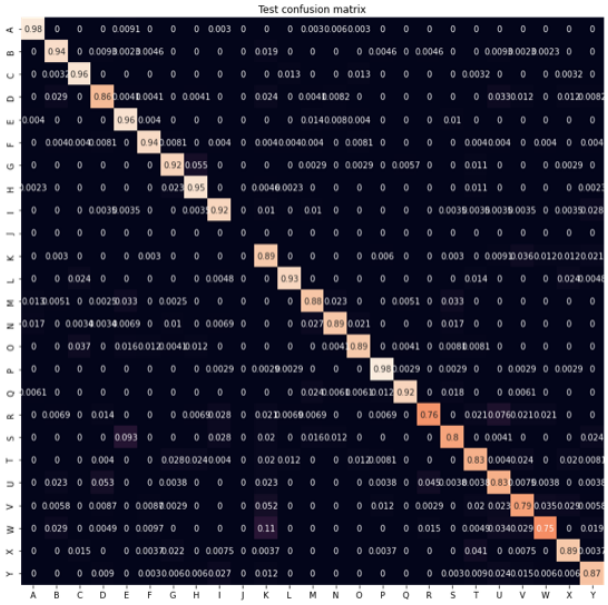


Figure 16: Dropout (10 ephoces with dropout rate = 0.5) (Acc = 89.8)

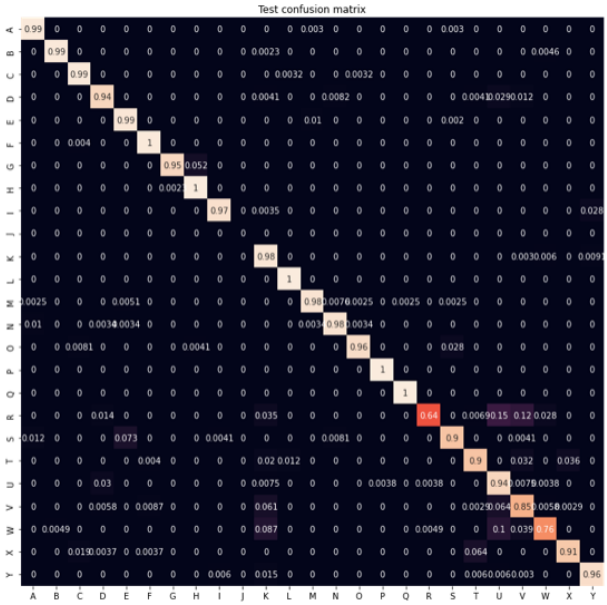


Figure 17: Dropout (6 ephoces with dropout rate = 0.17) (Acc = 92.1)

2.2.4 Batch Normalization

For this part I used 5 layers with batch normalization and there was a small improvement in accuracy.

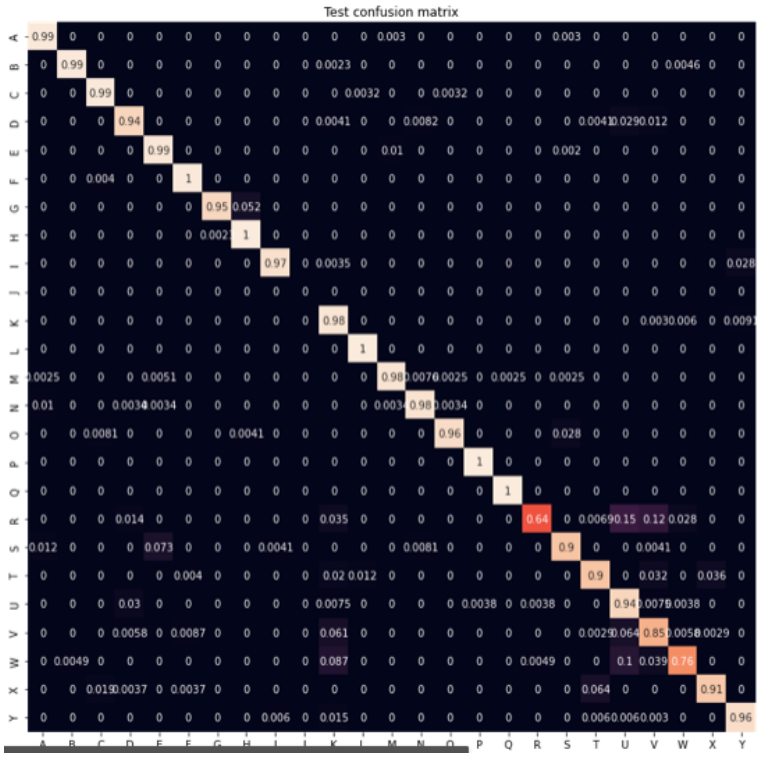


Figure 18: (Acc = 95.2)

2.2.5 Unbalanced data-set

For Label D, I had 176 samples in training data-set and I just kept 26 of them randomly. As you can see, error increased in label D. For solving the unbalanced data-set problem, we can repeat the available data or we can estimate its distribution and make more random samples from that. We can also use rotation or scaling approaches for that category images to increse their number.

2.2.6 Webcam

I also did this part in corporation with my friend Ali Arasteh. Although I trained and tested different networks and approaches, the result was not

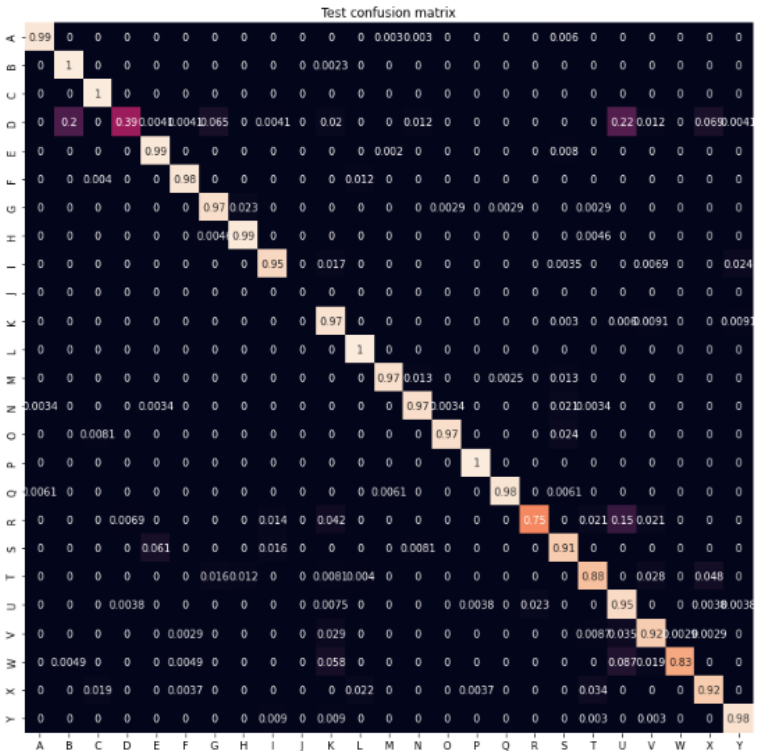


Figure 19: Unbalanced data-set for label 'D'

really perfect but for example I got the answer for L label. The problem here is that we can not normalize just one picture and we should have more images from that environment. Also as I overlooked the data-set, the images was really simple usually with solid background. We should have more powerful data-set for training.

3 Inverted Pendulum

I completed this part. All of the codes and results are available in this github reposiroty.