



به نام خدا



1928

K. N. Toosi University of Technology

دانشگاه صنعتی خواجه نصیرالدین طوسی

دانشکده برق

مبانی هوشند سازی

گزارش تمرین شماره 2

سجادرجبی باغستان

40005393

استاد : جناب دکتر مهدی علیاری

بهمن 1403

## فهرست مطالب

عنوان	شماره صفحه
سوالات.....	3
سوال اول.....	3
1.....	3
2.....	3
سوال دوم.....	5
سوال سوم.....	20
سوال چهارم.....	23

## سوال اول

1.

### حذف اطلاعات مهم توسط ReLU:

فعال ساز ReLU ممکن است تمام مقادیر منفی را به صفر تبدیل کند. در نتیجه، اگر اطلاعات مهمی در مقادیر منفی باشد، این اطلاعات از دست می‌روند. این می‌تواند باعث شود که مدل عملکرد بهینه‌ای نداشته باشد.

### اعوجاج خروجی سیگموئید:

خروجی سیگموئید با فرض اینکه ورودی آن می‌تواند مقادیر مثبت و منفی باشد طراحی شده است. اگر ورودی سیگموئید تنها غیرمنفی باشد (به دلیل ReLU در لایه قبلی)، احتمال تولید شده توسط سیگموئید ممکن است به طور غیرطبیعی به سمت 0/5 یا 1 متمایل شود. این می‌تواند عملکرد مدل را تحت تأثیر قرار دهد.

### عدم سازگاری معماری با توابع زیان:

در مسائل طبقه‌بندی دوکلاسه، معمولاً از تابع زیان باینری کراس انتروپی (BCE) استفاده می‌شود. این تابع به ورودی سیگموئید متکی است که مقادیر متقارن و با توزیع کامل از مثبت و منفی را می‌گیرد. اگر ورودی سیگموئید توسط ReLU محدود به مقادیر غیرمنفی شده باشد، محاسبات احتمال و گرادیان دچار اشکال خواهند شد.

2.

$$\text{grad}(\text{ELU}) = \begin{cases} 1 & x \geq 0 \\ \alpha e^x & x < 0 \end{cases}$$

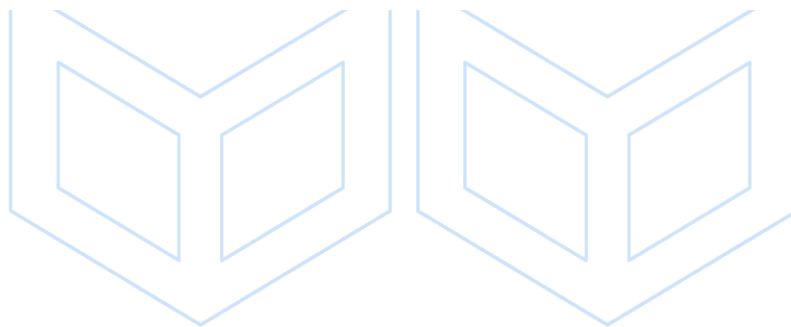
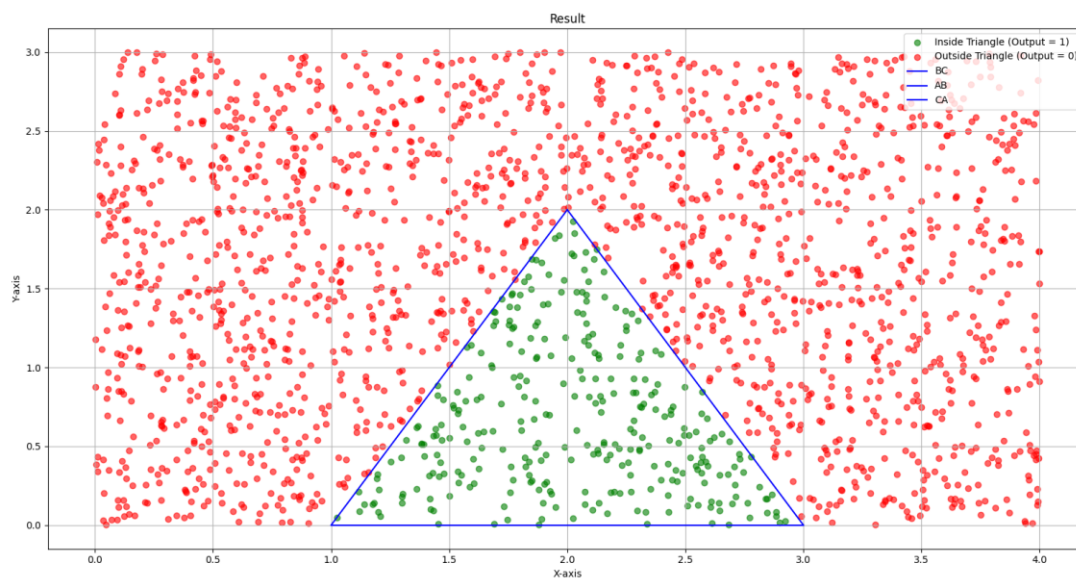
در ReLU، اگر ورودی  $x \leq 0$  باشد، گرادیان صفر است. این بدان معناست که نورون‌هایی که مقدار ورودی منفی دریافت می‌کنند، به طور کامل غیرفعال می‌شوند و در فرآیند یادگیری شرکت نمی‌کنند. این پدیده که به "مشکل مرگ نورون‌ها" (Dying ReLU Problem) معروف است، می‌تواند باعث شود بخش قابل توجهی از نورون‌ها در طول آموزش بی‌اثر شوند.

اما برای ELU، برای مقادیر  $x \leq 0$ ، گرادیان ELU برابر  $\alpha e^x$  است که هرگز صفر نمی‌شود.  
 (مگر اینکه  $\alpha = 0$  باشد). این ویژگی تضمین می‌کند که نورون‌ها حتی در نواحی منفی همچنان گرادیان غیرصفر دارند و در فرآیند یادگیری مشارکت می‌کنند.

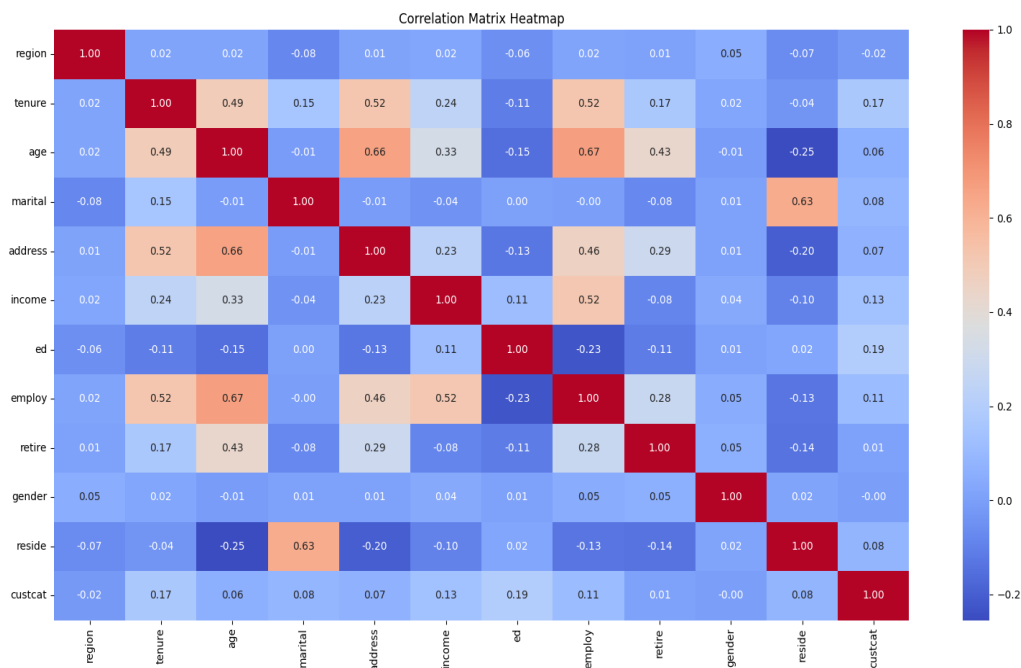
3.

برای این مسئله، یک تابع تعریف می‌کنیم که سه شرط را بررسی می‌کند که در واقع معدلات سه خط هستند .

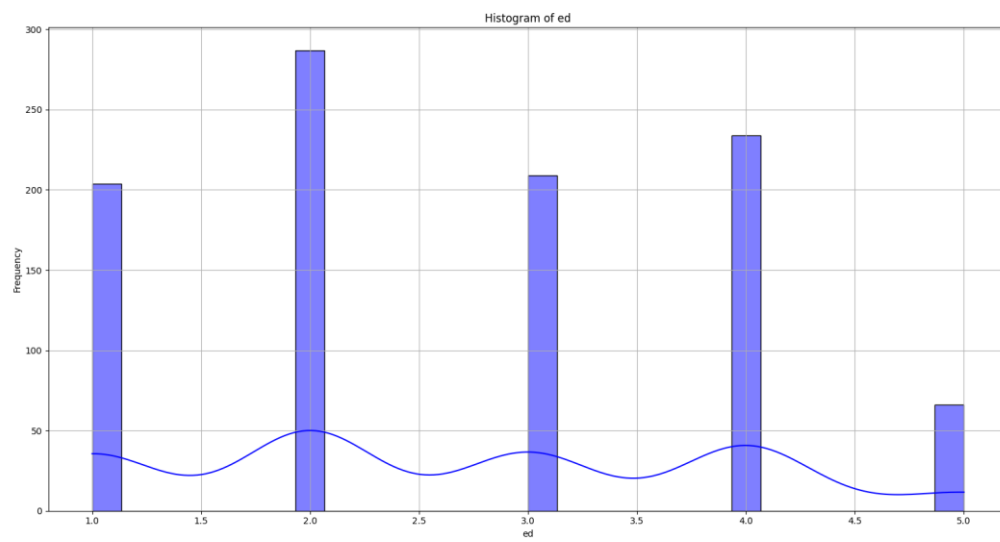
```
condition1 = y >= 0 # Above line BC
condition2 = y <= -2 * x + 6 # Below line AB
condition3 = y <= 2 * x - 2 # Below line CA
```

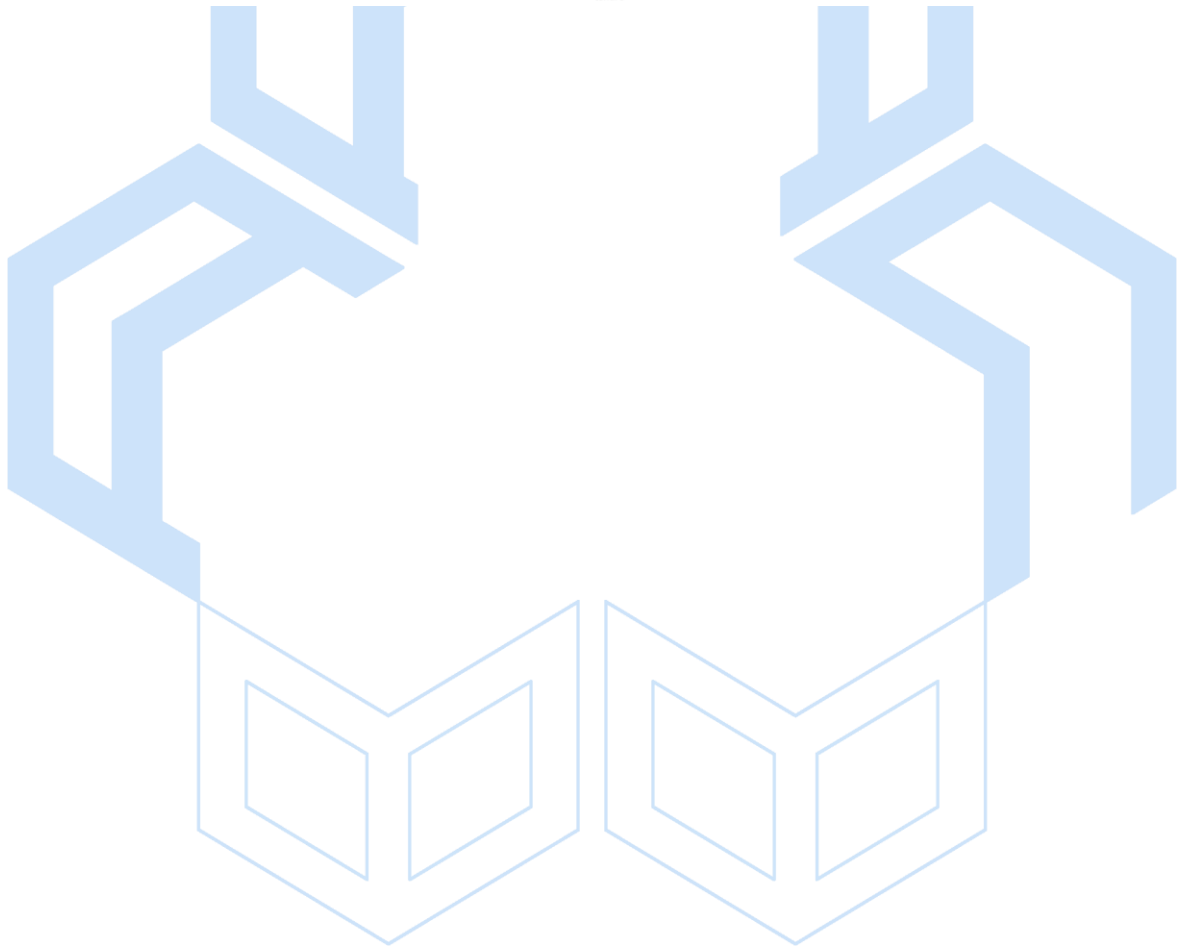
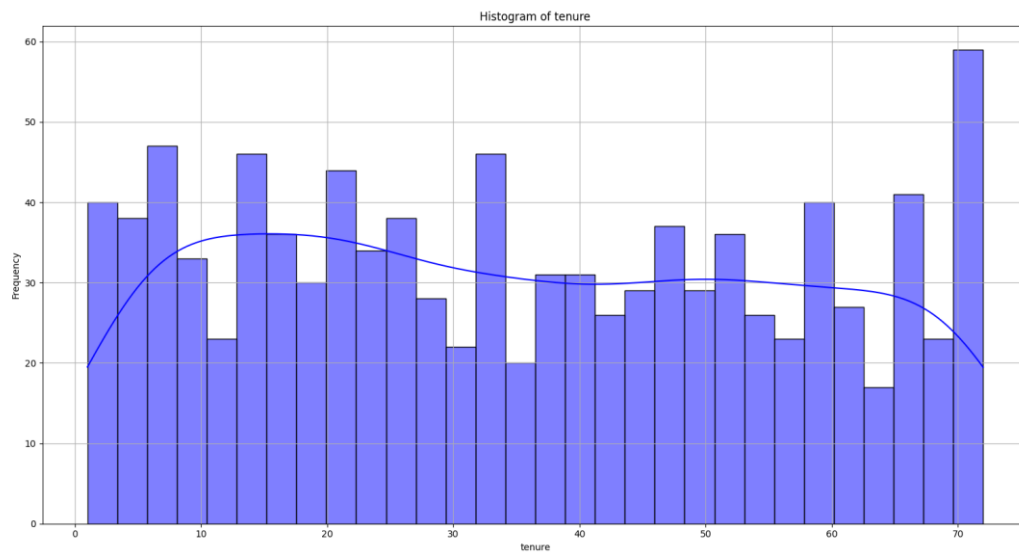


## سوال دوم



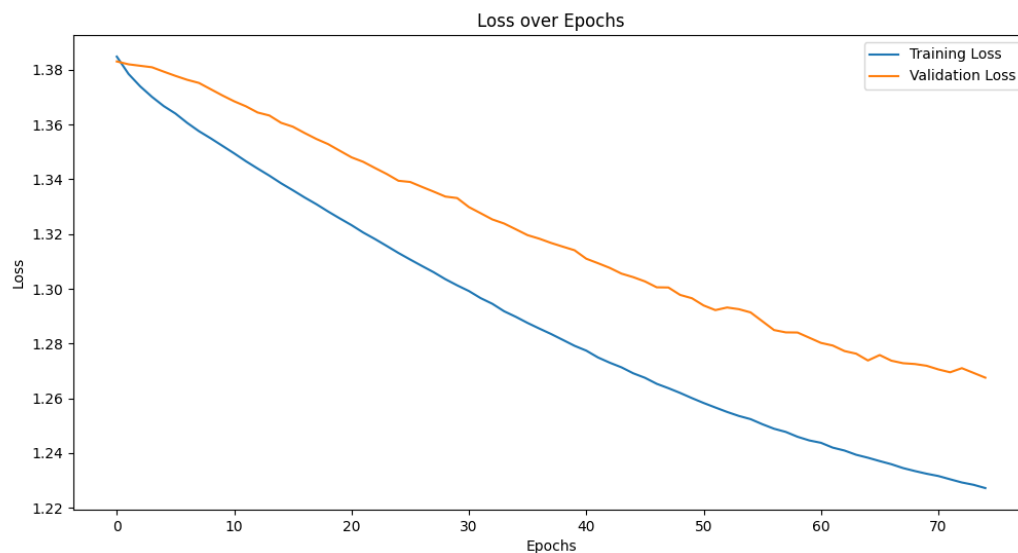
داده‌های موجود همبستگی زیادی با تارگت ما یعنی custcat ندارند و بیشترین همبستگی مربوط به داده ed است که برابر با ۰.۱۹ می‌باشد.





(2.4

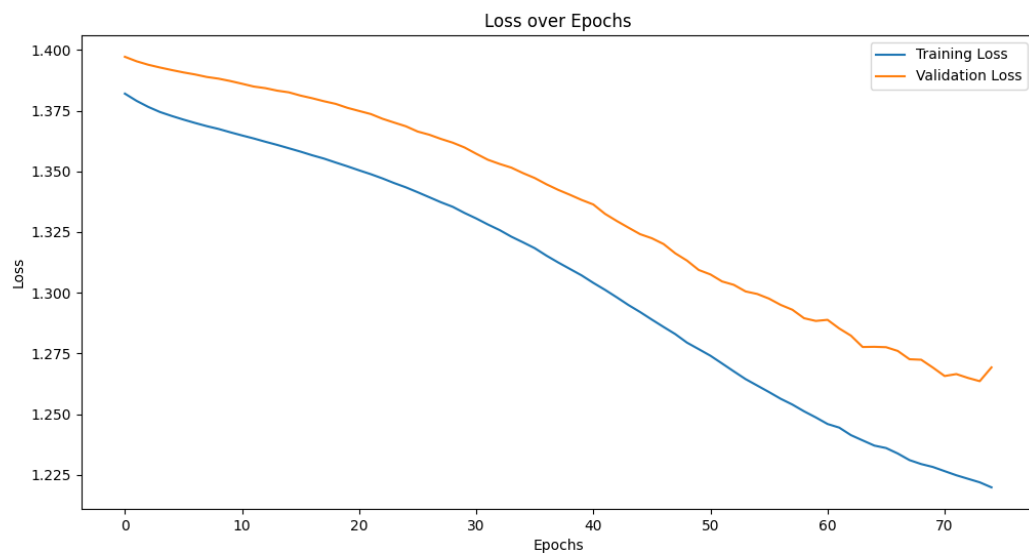
## مدل اول با 80 نرون



Test Accuracy: 0.38, Test Loss: 1.29

نتیجه به دست آمده (Test Accuracy) تقریباً برای تعداد نرون ها از 10 عدد تا 300 عدد یکسان بود تنها تفاوت اینها هزینه محاسباتی اضافه بود و مقدار Test Accuracy تقریباً بین 0.35 تا 0.41 بود. یکی دیگر از تفاوت تعداد نرون های بالا با تعداد پایین داشت این بود که مقدار loss با سرعت بیشتری کاهش پیدا میکرد ولی به هر حال در نتیجه تست تأثیری نداشت.

مدل دوم با 100 نورون در لایه اول و 50 نورون در لایه دوم

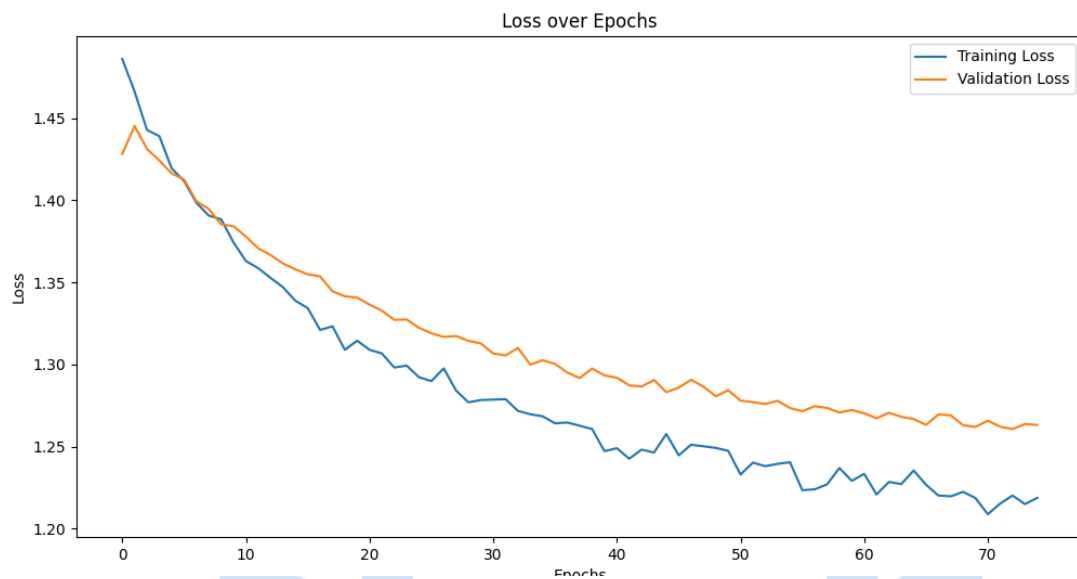


Test Accuracy: 0.38, Test Loss: 1.30

این تغییر در ساختار شبکه کمک خاصی به نتیجه نکرد و دقیقاً مثل شبکه قبلی نتیجه‌ها حدود 0.35 تا 0.41 است.



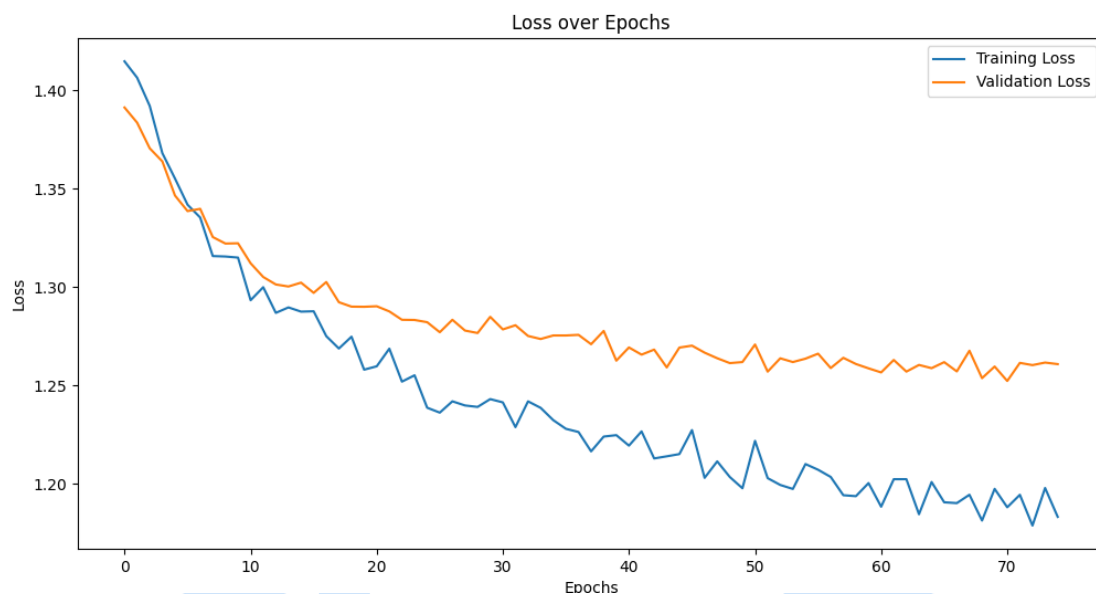
## مدل اول با 80 نورون با لایه نرمال سازی



Test Accuracy: 0.39, Test Loss: 1.31

با توجه به نمودار سرعت یادگیری مدل افزایش پیدا کرده که از فواید استفاده از لایه نرمال سازی است اما این تکنیک اثر خاصی در نتیجه تست نگذاشته است.

مدل دوم با 100 نورون در لایه اول و 50 نورون در لایه دوم با لایه نرمال سازی

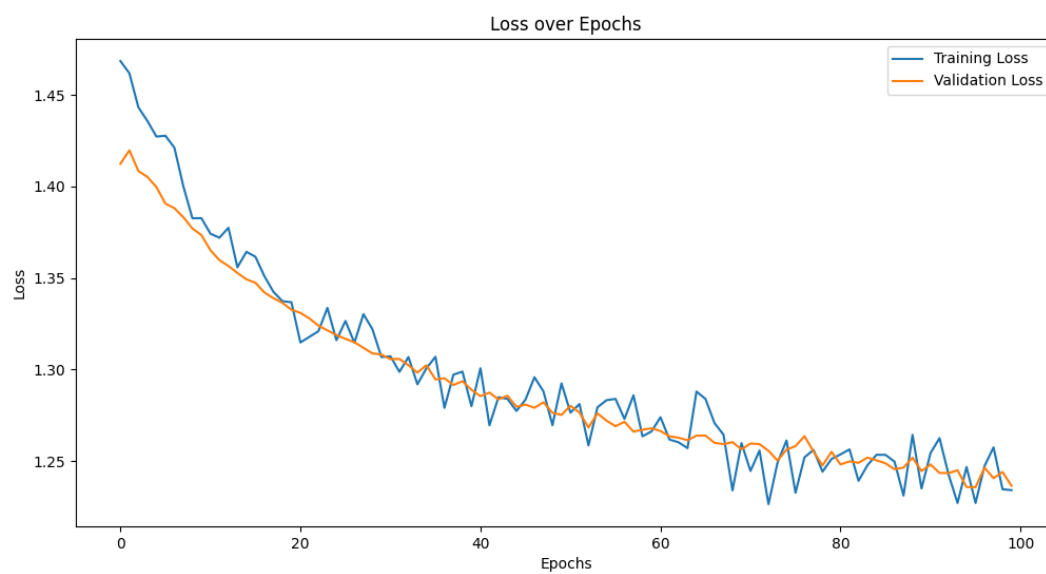


Test Accuracy: 0.38, Test Loss: 1.30

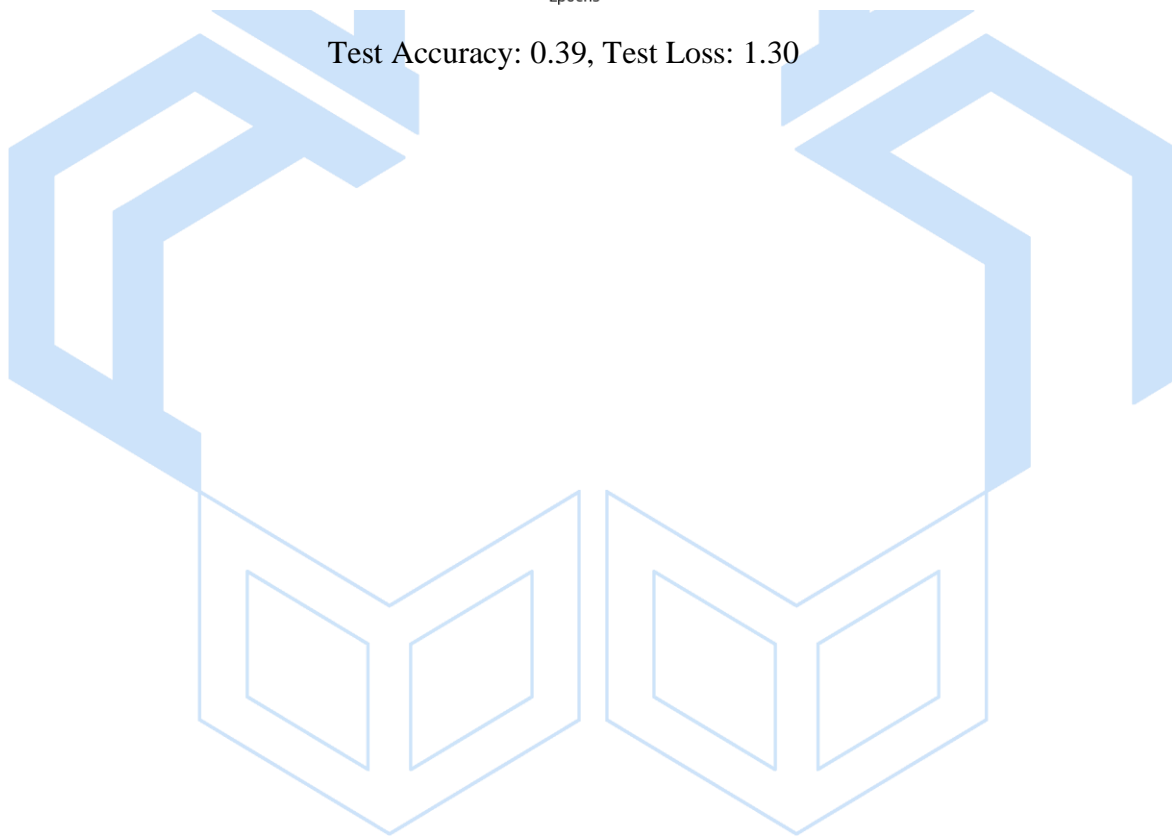
با توجه به نمودار سرعت یادگیری مدل افزایش پیدا کرده که از فواید استفاده از لایه نرمال سازی است اما این تکنیک اثر خاصی در نتیجه تست نگذاشته است.

اضافه کردن **Batch Normalization** اغلب ظرفیت مدل را برای یادگیری پیچیدگی های بیشتری از داده ها افزایش می دهد. این افزایش ظرفیت می تواند مدل را بیشتر در معرض **Overfitting** قرار دهد، به خصوص اگر از روش های منظم سازی (Regularization) مانند Dropout یا Weight Decay به اندازه کافی استفاده نکرده باشید.

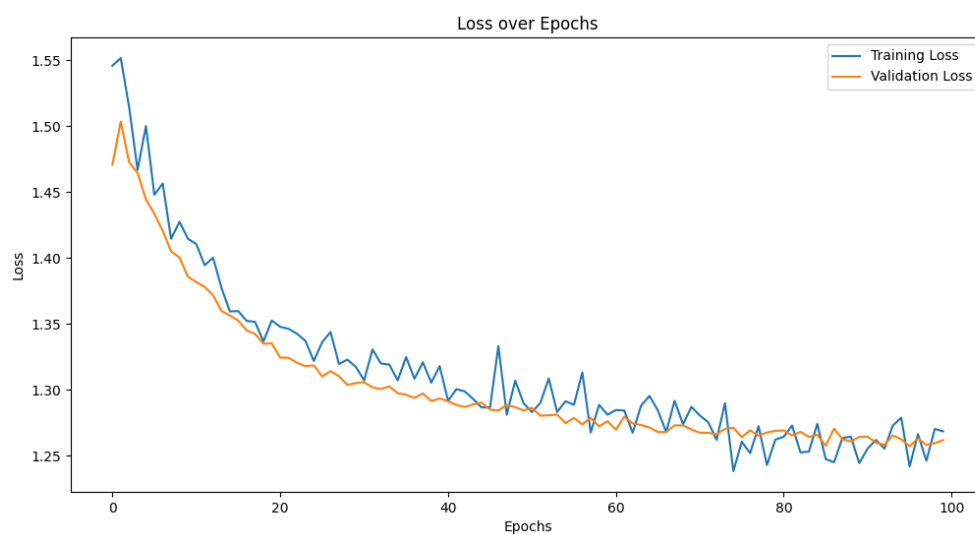
## مدل اول با 80 نورون با لایه نرمال سازی و dropout



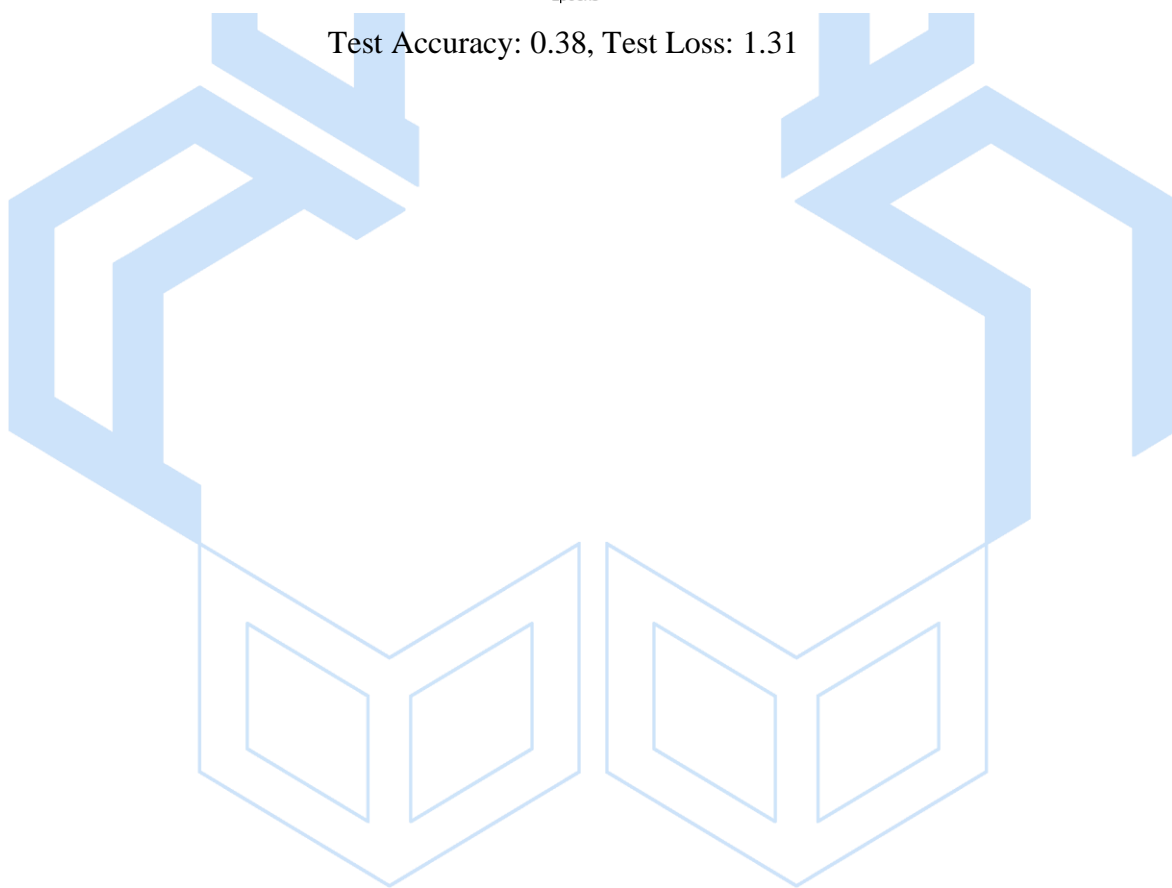
Test Accuracy: 0.39, Test Loss: 1.30



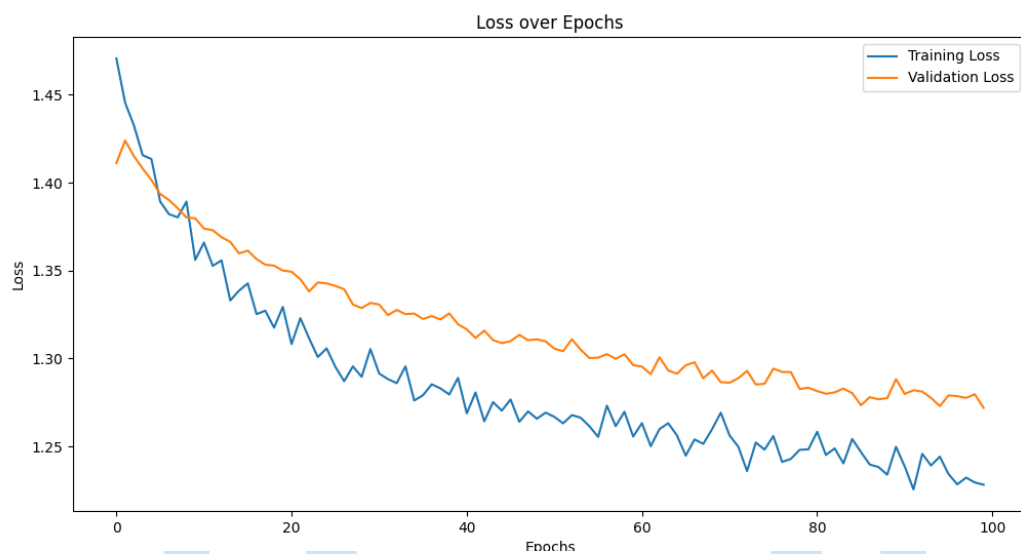
مدل دوم با 100 نورون در لایه اول و 50 نورون در لایه دوم با لایه نرمال سازی و dropout



Test Accuracy: 0.38, Test Loss: 1.31



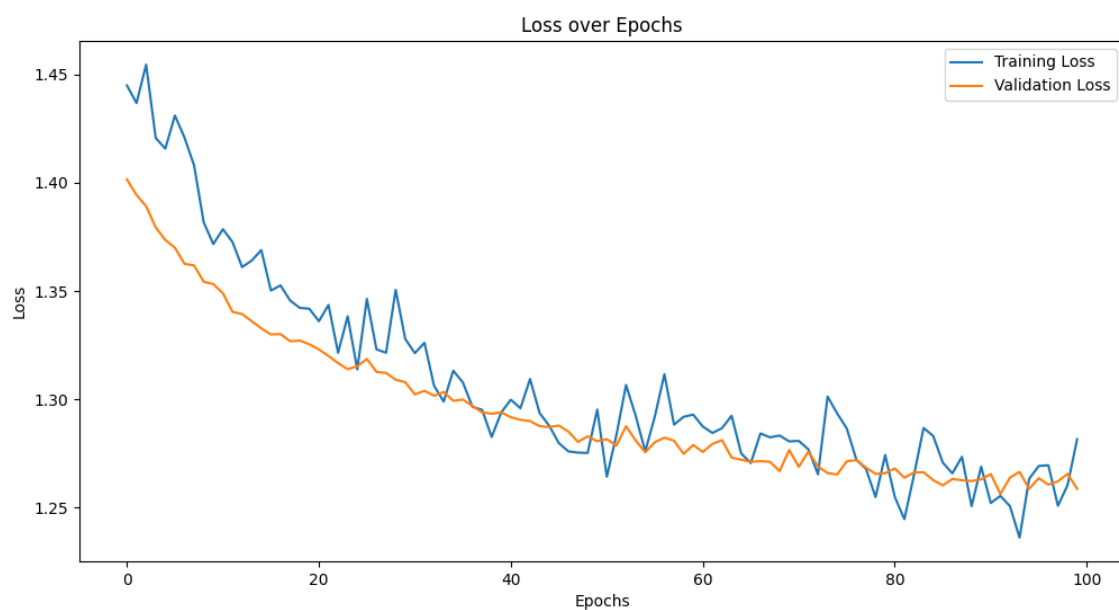
## مدل اول با 80 نورون با لایه نرمال سازی و dropout و L2 Regularization



Test Accuracy: 0.39, Test Loss: 1.30

مدل دوم با 100 نورون در لایه اول و 50 نورون در لایه دوم با لایه نرمال سازی و dropout و

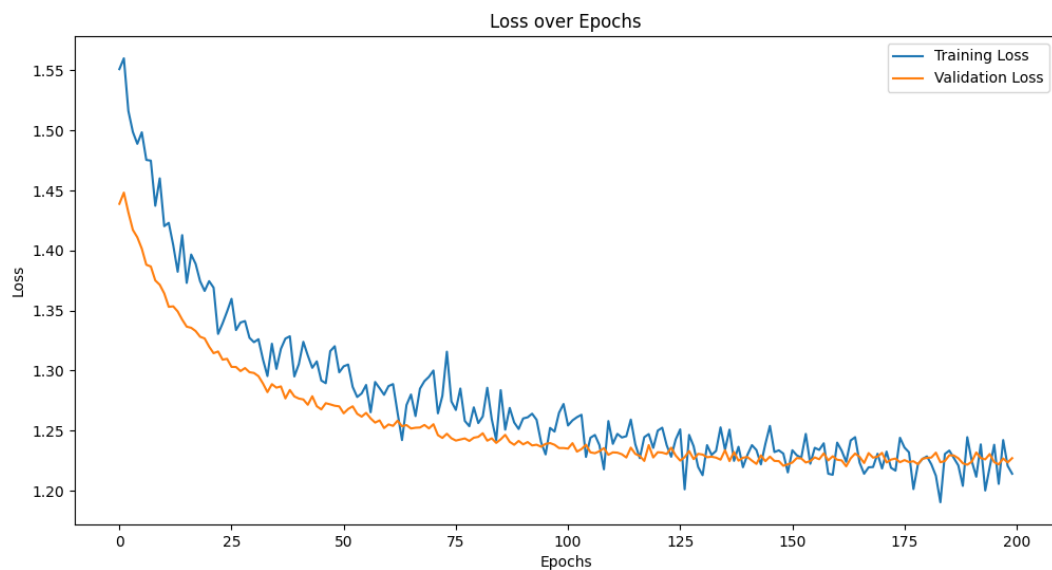
L2 Regularization



Test Accuracy: 0.38, Test Loss: 1.31

با تشکر.

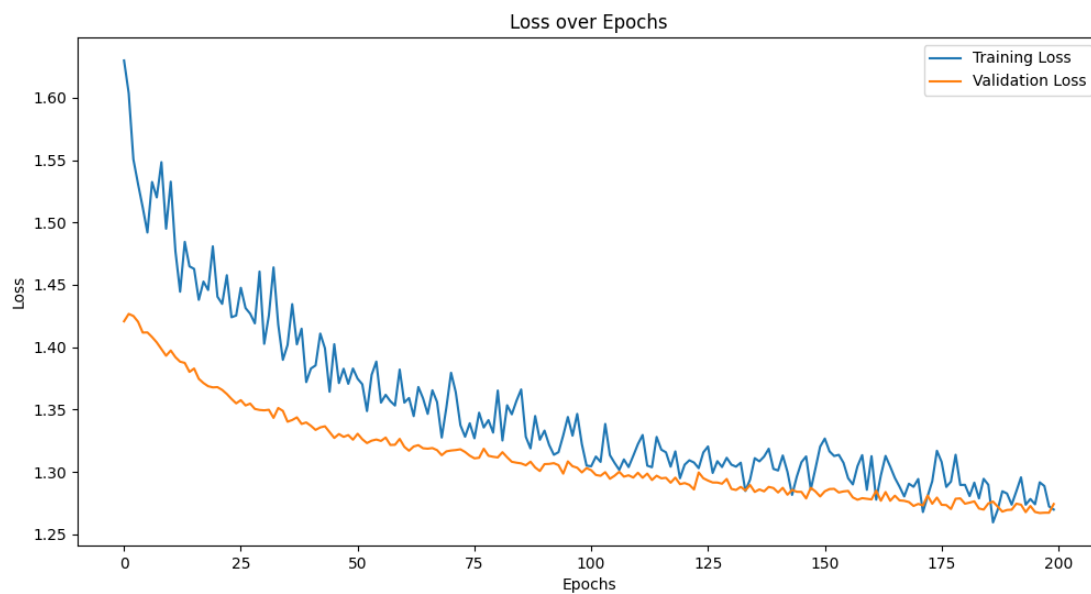
مدل اول با 80 نورون و بهینه سازی Adam با لایه نرمال سازی و dropout و L2 Regularization



Test Accuracy: 0.38, Test Loss: 1.30

مدل دوم با 100 نورون در لایه اول و 50 نورون در لایه دوم و بهینه ساز Adam با لایه نرمال سازی و

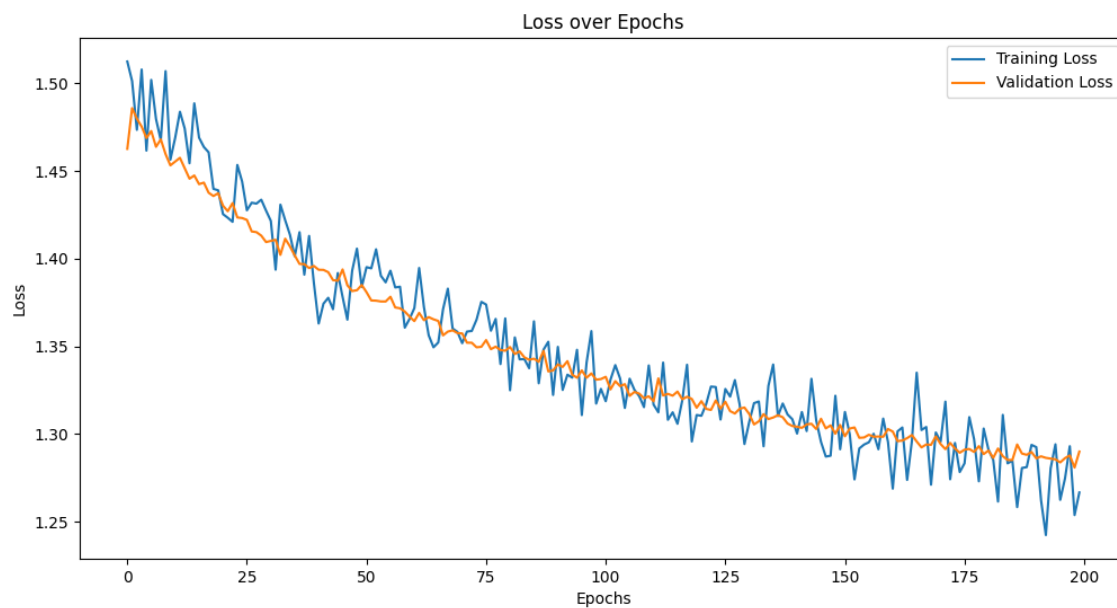
**L2 Regularization و dropout**



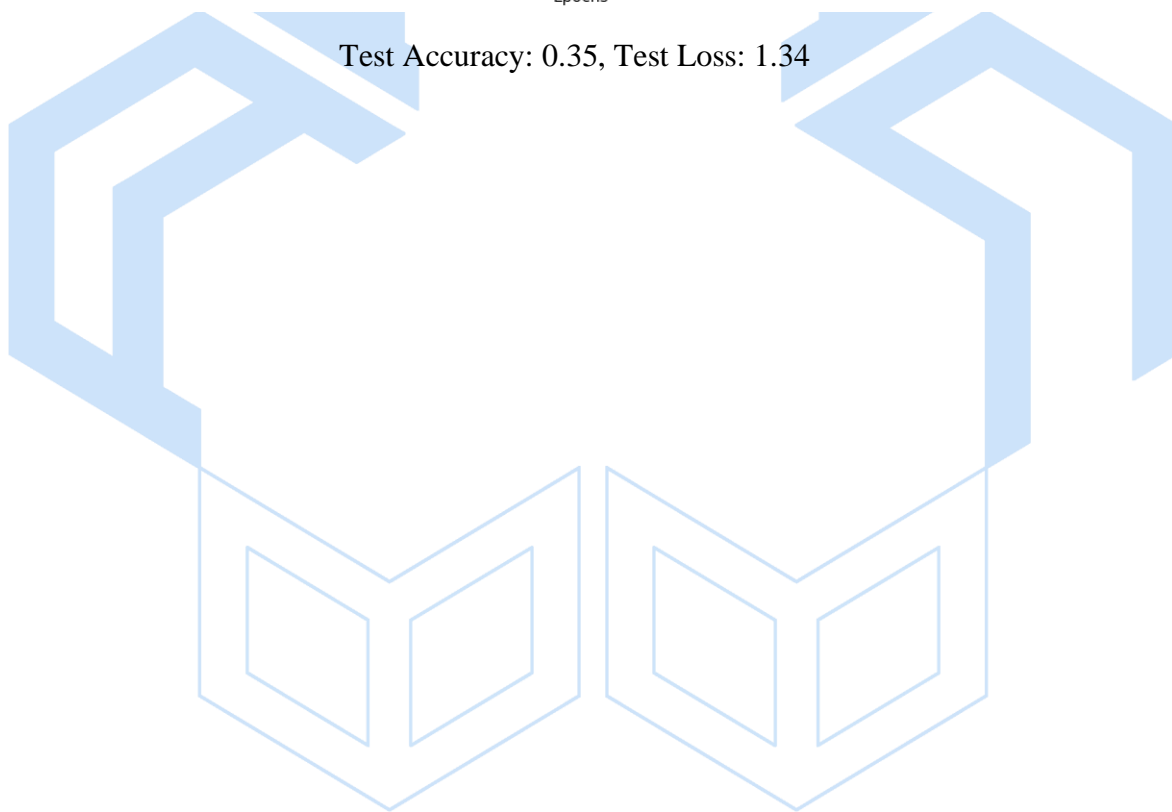
Test Accuracy: 0.34, Test Loss: 1.31



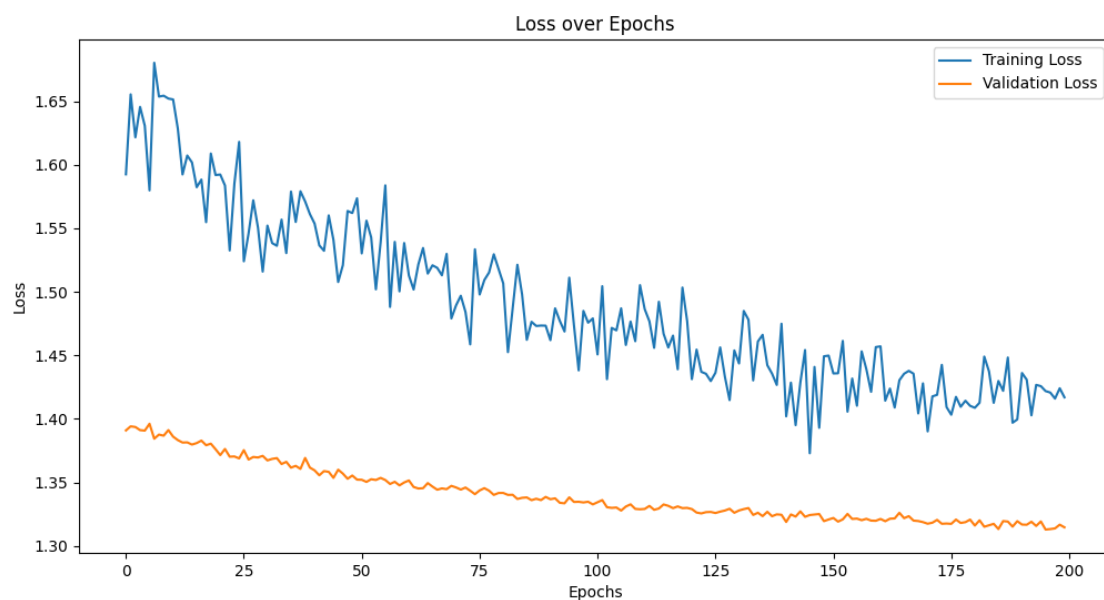
## مدل اول با 80 نورون و بهینه سازی Adadelta با لایه نرمال سازی و dropout و L2 Regularization



Test Accuracy: 0.35, Test Loss: 1.34



مدل دوم با 100 نورون در لایه اول و 50 نورون در لایه دوم و بهینه ساز Adadelta با لایه نرمال سازی و dropout و L2 Regularization



Test Accuracy: 0.35, Test Loss: 1.33

مدل اول با 80 نورون و نرمال ساز SGD با لایه نرمال سازی و dropout و L2 Regularization

Random Samples: Actual vs Predicted

Sample 1: Actual: 2, Predicted: 2

Sample 2: Actual: 2, Predicted: 2

Sample 3: Actual: 3, Predicted: 3

Sample 4: Actual: 0, Predicted: 2

Sample 5: Actual: 0, Predicted: 0

Sample 6: Actual: 1, Predicted: 0

Sample 7: Actual: 2, Predicted: 2

Sample 8: Actual: 0, Predicted: 0

Sample 9: Actual: 1, Predicted: 0

Sample 10: Actual: 2, Predicted: 1

مدل دوم با 100 نورون در لایه اول و 50 نورون در لایه دوم و نرمالساز SGD با لایه نرمال سازی و

**L2 Regularization و dropout**

Random Samples: Actual vs Predicted

Sample 1: Actual: 1, Predicted: 3

Sample 2: Actual: 2, Predicted: 2

Sample 3: Actual: 3, Predicted: 3

Sample 4: Actual: 3, Predicted: 0

Sample 5: Actual: 1, Predicted: 0

Sample 6: Actual: 3, Predicted: 1

Sample 7: Actual: 1, Predicted: 0

Sample 8: Actual: 0, Predicted: 0

Sample 9: Actual: 2, Predicted: 2

Sample 10: Actual: 2, Predicted: 3

## سوال سوم

تابع اول ابتدا ابعاد تصویر ورودی را استخراج می کند و سپس یک مقدار مشخص (مانند یک عامل یا آستانه) برای تشخیص سفید یا سیاه بودن هر پیکسل تعریف می کند. سپس مقدار RGB هر پیکسل به صورت جداگانه بررسی شده و شدت کلی نور (total intensity) آن محاسبه می شود. بر اساس آستانه مشخص شده، تعیین می شود که پیکسل مورد نظر سفید یا سیاه است و نتیجه به تصویر باینری جدیدی با نام `binary_representation` اضافه می شود.

روش جایگزین دیگری برای این کار به این صورت است:

ابتدا تصویر به مقیاس خاکستری (grayscale) تبدیل می شود و به جای مقادیر RGB، فقط یک مقدار شدت روشنایی برای هر پیکسل ذخیره می شود. سپس ماتریسی به ابعاد تصویر اصلی (به نام `binary_img`) ایجاد می شود که در آن پیکسل هایی با شدت روشنایی بیشتر از ۷۵ به مقدار `True` و کمتر از آن به مقدار `False` تغییر داده می شوند. در نهایت، با استفاده از تابع `astype(np.int8)` این مقادیر به ۱ و ۰ تبدیل می شوند.

در تابع دوم برای افزودن نویز به تصویر، می توان شدت نویز را با استفاده از متغیری به نام `noise_factor` تنظیم کرد. نویز به این صورت تولید می شود که برای هر پیکسل، عددی تصادفی در بازه ای بین `-noise_factor` و `noise_factor` انتخاب شده و به مقدار RGB آن پیکسل اضافه می شود.

Noise factor = 200

Matched Clean Image 1



Noisy Image 1



Noise factor = 1000

Matched Clean Image 1



Noisy Image 1

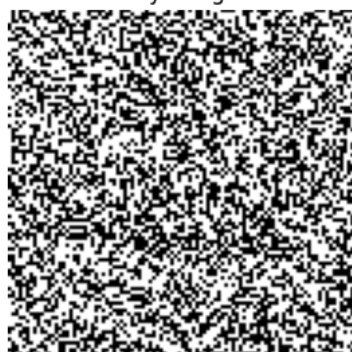


Noise factor = 2000

Matched Clean Image 4



Noisy Image 1



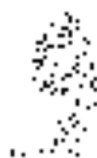
با نویز زیاد گاهی شبکه اشتباه حدس میزنه.

Noise factor = 500

Matched Clean Image 5



Missing Point 5



Noise factor = 1000

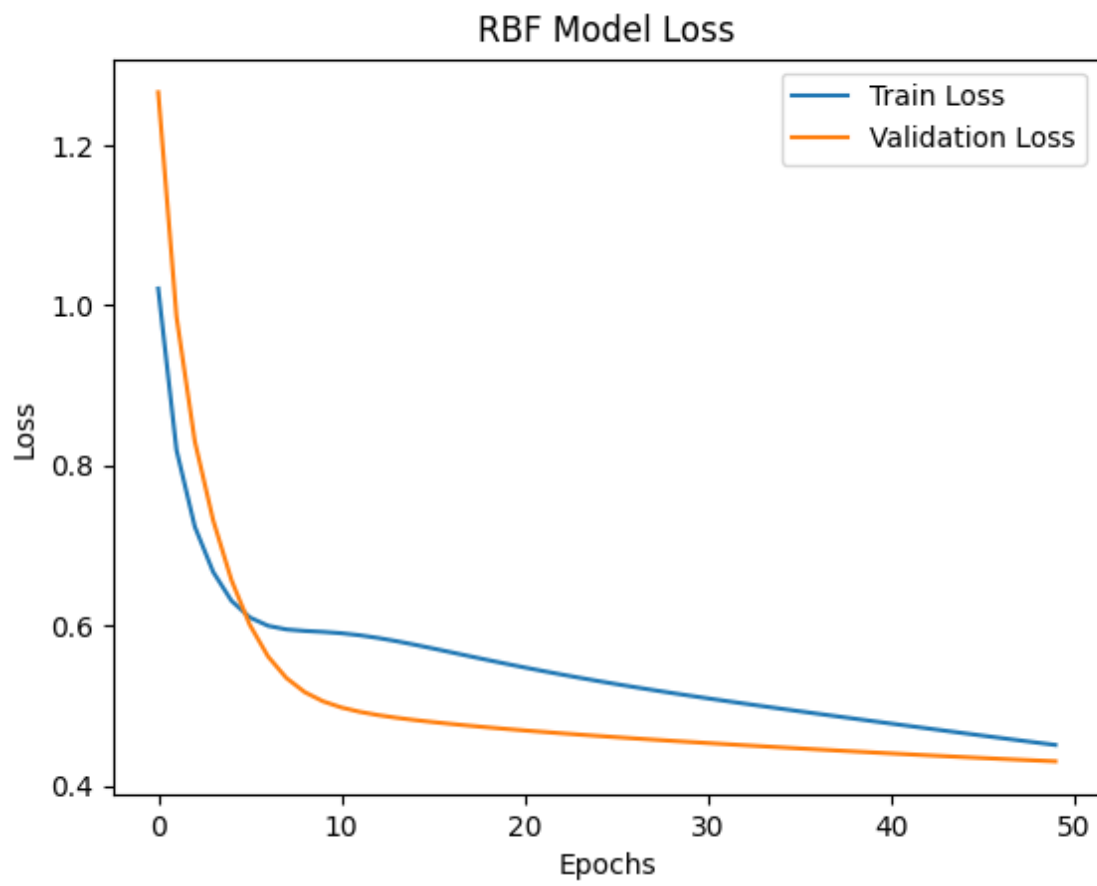
Matched Clean Image 4

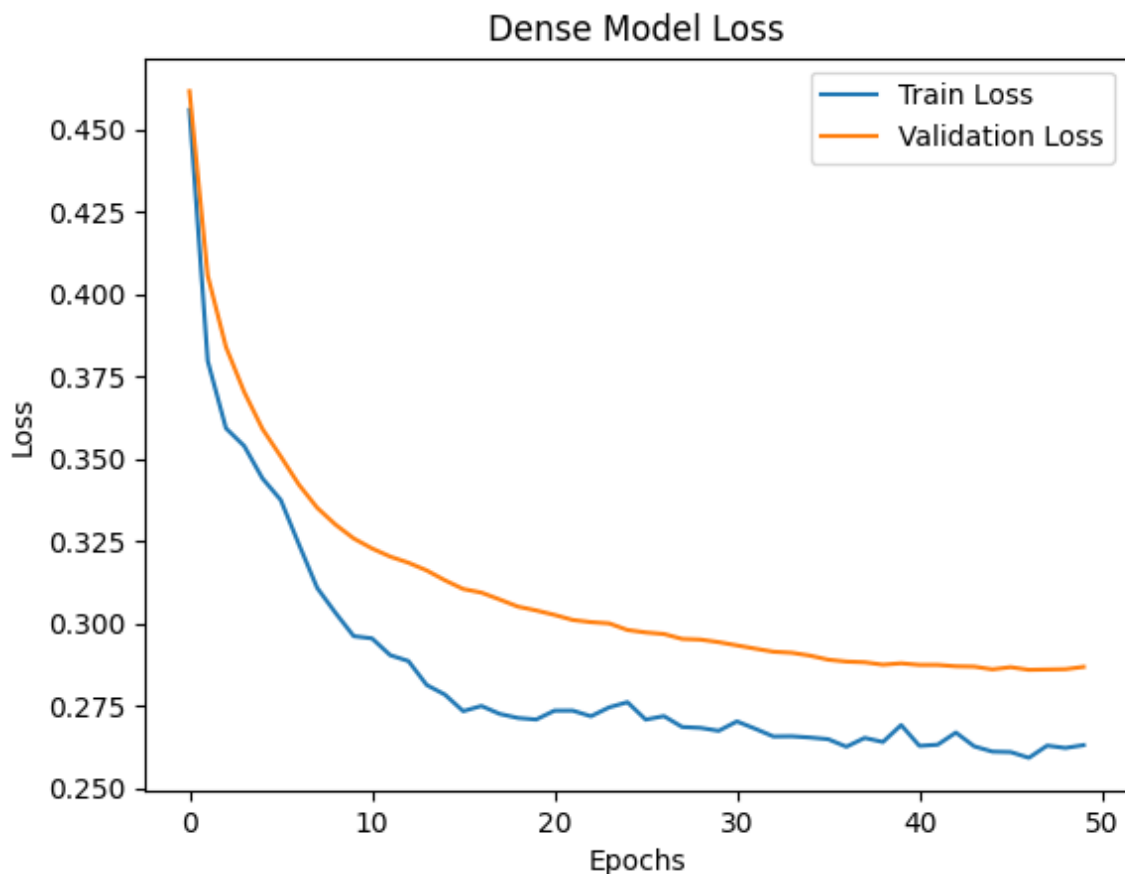


Missing Point 5



برای بهبود عملکرد شبکه در عکس‌های Missing Point، می‌توان از روش آگمنتیشن استفاده کنیم.





مدل Dense در مقایسه با مدل RBF عملکرد بهتری داشته و این موضوع از مقدار کمتر loss در این مدل مشخص است. این تفاوت نشان می‌دهد که مدل Dense توانایی بیشتری در حل مسئله مورد نظر دارد.

مدل RBF معمولاً زمانی نتایج بهتری ارائه می‌دهد که داده‌ها دارای پراکندگی خوشه‌ای باشند. اما در این مسئله، روابط داده‌ها پیچیده و غیرخطی بوده و با هدف نهایی ارتباطات غیرساده‌ای دارند. به همین دلیل، مدل Dense با ساختار انعطاف‌پذیرتر و توانایی بهتر در یادگیری روابط پیچیده، انتخاب مناسب‌تری برای این مسئله است.