

**درس رایانش عصبی و یادگیری عمیق**

**استاد صفابخش**

**نیما پری فرد**

**402131017**

فهرست تمرین شماره 1

[تمرین 1 3](#_Toc162211479)

[1-a: 3](#_Toc162211480)

[1-b: 4](#_Toc162211481)

[1-c: 7](#_Toc162211482)

[تمرین 2 9](#_Toc162211483)

[2-a: 10](#_Toc162211484)

[2-b: 15](#_Toc162211485)

[2-c: 19](#_Toc162211486)

[تمرین 3 24](#_Toc162211487)

[3-a: 24](#_Toc162211488)

[3-b: 24](#_Toc162211489)

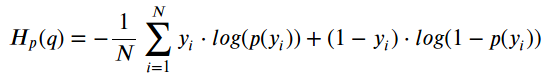
[3-c: 26](#_Toc162211490)

# تمرین 1

امکان ندارد که خطا صفر شود اثبات آن بسیار ساده است کافی است که به فرمول softmax و BCE نگاهی بیاندازیم.

A mathematical equation with black text

Description automatically generated



برای اینکه خطا این جا صفر شود بایستی که خروجی لایه softmax 1 شود.

A graph of a number of different colored lines

Description automatically generated

و این خاصیت لایه softmax است که خروجی عددی بین 0 تا 1 است که خورد 0 و 1 را شامل نمی شود برای همین امکان ندارد لاس صفر شود.

# تمرین 2

**اندازه دسته و عوامل موثر بر انتخاب آن**

اندازه دسته در فرآیند آموزش شبکه عصبی، به تعداد داده های نمونه آموزش در iteration در طول فرآیند گرادیان کاهشی اشاره دارد.

بجای اینکه بیام و برای کل داده آموزش گرادیان حساب کنیم. می آییم و برای زیر مجموعه ای از داده های حساب می کنیم.

عوامل موثر در انتخاب سایز دسته:

* منابع محاسباتی: حافظه و قدرت پردازش موجود به طور قابل توجهی بر انتخاب اندازه دسته تأثیر می‌گذارد. دسته‌های بزرگتر نیاز به حافظه بیشتری دارند، بنابراین اگر سیستم منابع کافی را نداشته باشد، استفاده از دسته‌های کوچک ضروری می‌شود.
* پیچیدگی مدل: مدل‌های پیچیده با پارامترهای زیاد ممکن است نیازمند اندازه دسته کوچک‌تری باشند تا در طول آموزش در حافظه جا بگیرند. مدل‌های ساده می‌توانند اندازه دسته بزرگ‌تری را به طور موثر تر مدیریت کنند.
* اندازه داده‌های آموزش: اندازه مجموعه داده نقش حیاتی را ایفا می‌کند. مجموعه داده‌های کوچک از دسته‌های بزرگتر بهره می‌برند، در حالی که مجموعه داده‌های بزرگ همچنان تنوع کافی برای آموزش با دسته‌های کوچک را فراهم می‌کنند.
* نرخ یادگیری: اندازه دسته تأثیری بر نرخ یادگیری دارد. دسته‌های کوچک‌تر ممکن است نیاز به نرخ یادگیری کوچکتری داشته باشند تا جلوی بیش از اندازه حرکت کردن را بگیرند، در حالی که دسته‌های بزرگتر می‌توانند نرخ‌های یادگیری بالاتر را تحمل کنند.
* تعمیم: دسته‌های کوچک‌تر نویز بیشتری ایجاد می‌کنند که می‌تواند تعمیم را بهبود بخشد. با این حال، دسته‌های بزرگتر ممکن است به سرعت همگرایی داشته باشند.
* نزول گرادیان تصادفی در مقابل نزول گرادیان قطعی: دسته‌های کوچک‌تر منجر به نزول گرادیان تصادفی می‌شوند، در حالی که دسته‌های بزرگتر شبیه نزول گرادیان قطعی هستند. انتخاب بستگی به هدف بهینه‌سازی دارد.

انتخاب اندازه دسته یک پارامتر مهم در آموزش شبکه‌های عصبی است. بیایید به بررسی انتخاب بهینه اندازه دسته بپردازیم:

* حالت دسته: در این حالت، اندازه دسته برابر با اندازه کل مجموعه داده است، که هر دوره با هر بار تکرار معادل یک دوره است. این رویکرد برای مجموعه‌های آموزش کوچک مناسب است (زمانی که m کمتر از ۲۰۰ است).
* حالت مینی-دسته: در اینجا، اندازه دسته بیشتر از یک و کمتر از اندازه کل مجموعه داده است. اندازه‌های معمول مینی-دسته عبارتند از ۶۴، ۱۲۸، ۲۵۶ یا ۵۱۲. نزول گرادیان مینی-دسته نسبت به حالت دسته سریعتر یاد می‌گیرد.
* حالت تصادفی: در این حالت، اندازه دسته برابر با یک است. بعد از هر نمونه، گرادیان و پارامترهای شبکه عصبی به‌روزرسانی می‌شوند. اگرچه این رویکرد از تسریع‌های برداری جلوگیری می‌کند، اما ممکن است محاسباتی گرانبها باشد.

در مورد چالش گفته شده باید از نوع سوم استفاده کنیم یعنی از stochastic gradient decent استفاده کنیم.

یک راه دیگر این است که از شبکه های عصبی دیگر برای کوچک کردن سایز داده استفاده کنیم به طور مثال می توان از autoencoder استفاده کرد.

<https://medium.com/mini-distill/effect-of-batch-size-on-training-dynamics-21c14f7a716e>

# تمرین 3

در تئوری 3.2 مرجعی که اوردم اثبات کرده است که چگونه یک تابع محدود خطی درون یک ناحیه محدب محلی توسط یک RELU DNN ساخته یا نمایش داده می شود.

مراجع:

[machine learning - Can every piecewise linear function be exactly realized as a neural network? - Mathematics Stack Exchange](https://math.stackexchange.com/questions/4173446/can-every-piecewise-linear-function-be-exactly-realized-as-a-neural-network)

ReLU Deep Neural Networks and Linear Finite Elements

[1807.03973.pdf (arxiv.org)](https://arxiv.org/pdf/1807.03973.pdf)

# تمرین 4

تقارن در شبکه های عصبی به معنی این است که مجموعه ای از پارامتر ها یا نورون نقش یکسان یا غیر قابل تغییری داشته باشند.

ما تقارن در داده هم داریم که به این معنی است شبکه با داده های یکسان رفتار مشابه ای داشته باشد برای مثال عکس های گربه را دید با همه رفتار یکسان داشته باشد و بگوید عکس گربه است ما این تقارن را دوست داریم و باعث افزایش عملکرد شبکه و مقاوم پذیری آن می شود. فکر می کنم منظور بیش تر این سوال تقارن که در داخل شبکه عصبی که انواع آن ها در موارد زیر نوشتم.

انواع تقارن در شبکه عصبی:

1. تقارن در پارامتر ها: تقارن در پارامتر ها به پارامتر های قابل یادگیری شامل وزن ها و بایاس ها در شبکه عصبی کاربرد دارد. این تقارن زمانی اتفاق می افتد که با پارامتر های مختلف پیش بینی یکسانی برای ورودی های یکسان دارند.
2. تقارن در نمایش ها: این مورد به این اشاره دارد به نمایش داخلی از داده درون مدل های مختلف. فرض کنید مدل دارای معماری یکسانی است. در واقع این تقارن به این اشاره دارد که چگونه معماری تاثیر دارد بر نحوه نمایش. با محاسبه گروه های تقارن بنیادین که به ان interwiner groups گفته می شود ما می تونیم این نوع تقارن را کشف کنیم.
3. تقارن در تابع فعالیت: این تقارن به این اشاره دارد که به اعمال تابع فعالیت که به نورون های یکسان در سرتاسر شبکه.
4. تقارن permutation : این تقارن به شرایطی اشاره دارد که درجه نورون ها تاثیری در رفتار شبکه ندارد.

روش های شکستن تقارن در شبکه عصبی:

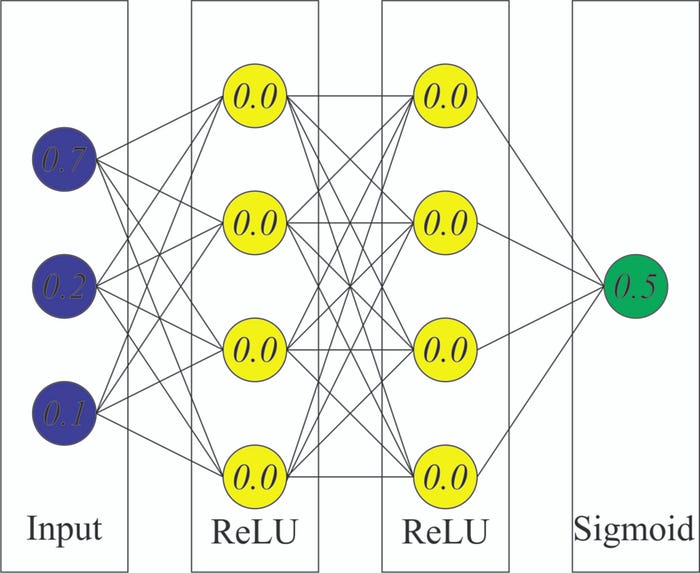
در اینجا چند مورد از روش های شکستن تقارن در شبکه عصبی می نویسم.

1. استفاده از روش های مقدار دهی اولیه مانند Xavier
2. استفاده از تابع فعالیت های متفاوت
3. استفاده از فعال سازی تصادفی نورون یعنی همان Dropout
4. استفاده از معماری شبکه متفاوت
5. Wight regularization
6. Data augmentation

مثالی از تقارن را می توان شبکه عصبی که وزن های ابتدایی آن همه صفر هستند در نظر گرفت، زمانی که عملیات backpropagation رخ می دهد گرادیان یکسان به عقب باز می گردد و همه وزن های یک سان به روز رسانی می شوند که باعث می شود، همه وزن های دوباره یکسان شوند به این مثال تقارن گفته می شود.

این مورد باعث می شود که تمام نورون ها یک چیز یاد بگیرند و این برای شبکه عصبی ما مناسب نیست.

حال برای حل مشکل این مثال می توانیم از مقداردهی اولیه رندم استفاده کنیم. به طوری که در این گرادیان یکسان نیست و وزن های مقدار یکتای خود را دارند در این جا ما با این تکنیک تقارن را شکاندیم.



پیاده سازی نمونه تقارن که بالا ذکر کردم.

A screen shot of a computer program

Description automatically generated

معماری نمونه که بالا گزاشتم فقط نمونه است و کاربرد دیگری ندارد

A screenshot of a computer program

Description automatically generated

در بالا اول وزن های شبکه را با مقدار اولیه صفر گزاشتم تا این وزن ها آموزش ببیند

A screen shot of a computer

Description automatically generated

در عکس صفح بعد مشاهده می کنید که بعد از آموزش وزن ها تغییر نکرده و همان صفر باقی مانده اند واضح که همه نورون یک کار یکسان انجام می دهند و تقارن در شبکه وجود دارد.

A computer screen shot of a program code

Description automatically generated

A computer screen shot of a program

Description automatically generated

در تابع بالا مقدار وزن ها تصادفی مقداردهی اولیه انجام دادم و برای آموزش در قیمت پایین مشاده می کنید که شبکه آموزش دیده و نورون با هم متفاوت هستند و هر کدام یک کاری را انجلم می دهند.

A computer screen shot of a program

Description automatically generated

مراجع:

On the Symmetries of Deep Learning Models and their Internal Representations

[2205.14258v1.pdf (arxiv.org)](https://arxiv.org/pdf/2205.14258v1.pdf)

<https://ml4physicalsciences.github.io/2021/files/NeurIPS_ML4PS_2021_117.pdf>

<https://ieeexplore.ieee.org/document/287176/>

# تمرین 5

A screen shot of a computer screen

Description automatically generated

## 5-a:

A screenshot of a computer program

Description automatically generated

A computer screen shot of a program code

Description automatically generated

A computer screen shot of a code

Description automatically generated

A computer screen shot of text

Description automatically generated

## 5-b:



A screen shot of a computer program

Description automatically generated

A storm clouds over an island

Description automatically generated

A storm clouds over a body of water

Description automatically generated

A graph of training and validation

Description automatically generated

## 5-c:

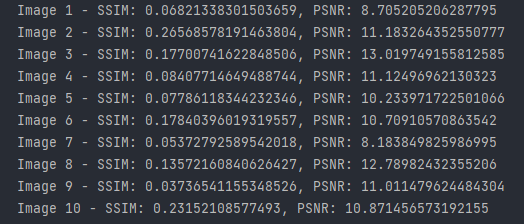
A screen shot of a computer program

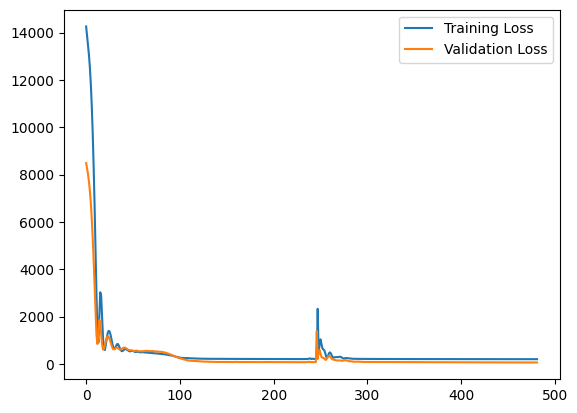
Description automatically generated

A screen shot of a computer

Description automatically generated

## 5-d:





A screenshot of a computer screen

Description automatically generated

A computer screen shot of text

Description automatically generated

A screen shot of a computer code

Description automatically generated



A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

A person holding shoes in their hand

Description automatically generated

A person holding a pair of shoes

Description automatically generated