

به نام خدا



دانشکده مهندسی کامپیوتر

تمرینات درس یادگیری عمیق

پاسخنامه تمرین سری چهارم

دکتر محمدرضا محمدی

دانشجو: مجتبی نافذ

96431335

نیمسال دوم ۹۹-۱۰۰

سوال اول:

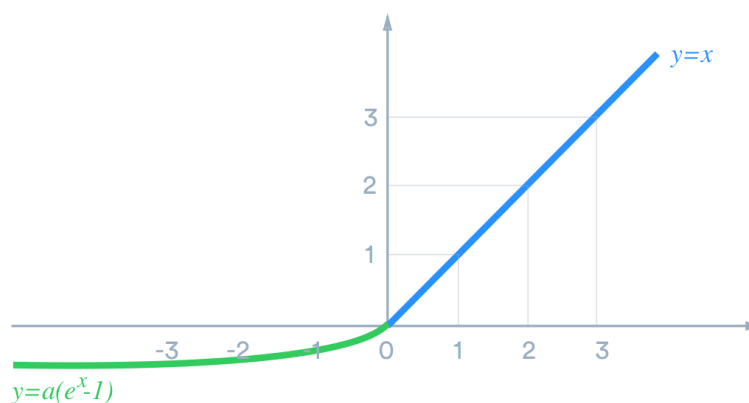
۱. الف) توابع فعال‌سازی ReLU، Leaky ReLU و ELU را مقایسه کنید. (از نظر پیچیدگی محاسباتی، معایب و مزایا)
- ب) توابع فعال‌سازی Mish و Swish را مطالعه کنید و به صورت مختصر با توابع فعال‌سازی قسمت قبل مقایسه کنید.
- پ) مشتق توابع ReLU، Leaky ReLU، ELU، Mish و Swish را به صورت دستی محاسبه کنید و معادله آنها را یادداشت کنید.
- ت) مشتق‌های محاسبه شده در قسمت قبل را ترسیم کنید (با استفاده از `pyplot.plot`) و مقایسه کنید.

الف) Relu: از مزایای آن : ۱- پیچیدگی ریاضیاتی ندارد و محاسباتی سبک نقطه قوت آن است. و در نتیجه سرعت آموزش مدل و اجرا بالاست و زمان کمی نیاز دارد ۲- سرعت همگرایی بالاست چون خطی است و در نتیجه شیب نه اشباع میشود و نه مسطح و پدیده ی محو شدن گرادیان هم مطرح نیست(البته فقط در سمت مثبت ها) ۳- به ندرت این تابع فعال میشود چون خیلی از خروجی ها منفی اند ۴- چون خیلی از نورون ها خروجی منفی میدهند و فعال نمی شوند باعث میشود داده ها به صورت پراکنده فعال شوند و در نتیجه در مدل spares یا پراکنده احتمال overfit کردن بسیار کم تر است.

از معایب : صفر بودن مشتق در خروجی های منفی است که باعث میشود شبکه حتی در صورت خطا آپدیت نشود.

تابع **Leaky ReLU** مزایا: باز هم پیچیدگی ریاضی زیادی ندارد اما از **ReLU** پیچیدگی آن بیشتر است ۲- در صورت خطا در خروجی های منفی هم آپدیت می شویم. و مشکل **ReLU** را تا حدی برطرف میکند. اما دقت کنید که همیشه به **ReLU** برتری ندارد و بستگی به داده ها دارد.

تابع **ELU** یا همان **Exponential Linear** یک شیب کم برای مقادیر منفی دارد و به جای یک خط صاف یک منحنی لگاریتمی برای مقادیر منفی دارد.



مزیت: برای مقادیر منفی مشکل صفر بودن گرادیان را ندارد و توقف نمی‌کند

مشکل: در کل در مقادیر منفی بزرگ گرادیان اشباع میشود. به لحاظ محاسباتی در مقادیر منفی محاسبات ریاضی به نسبت

زیادی دارد

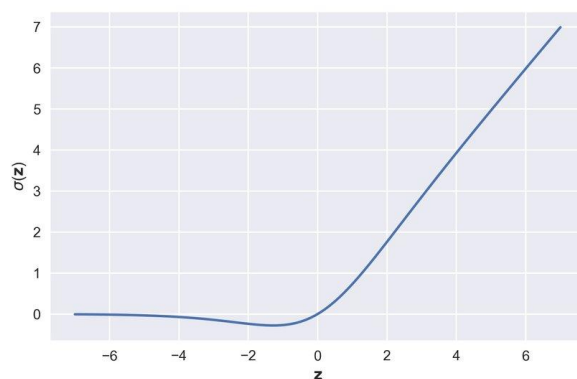
(ب)

Swish:

$$f(x) = x * \sigma(\beta x)$$

بتا یک پارامتر ثابت یا trainable است

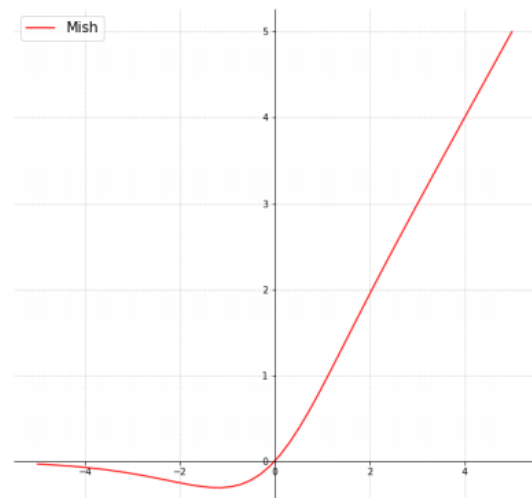
در شبکه های عمیق دقت بیشتری نسبت به ReLu دارد. ب



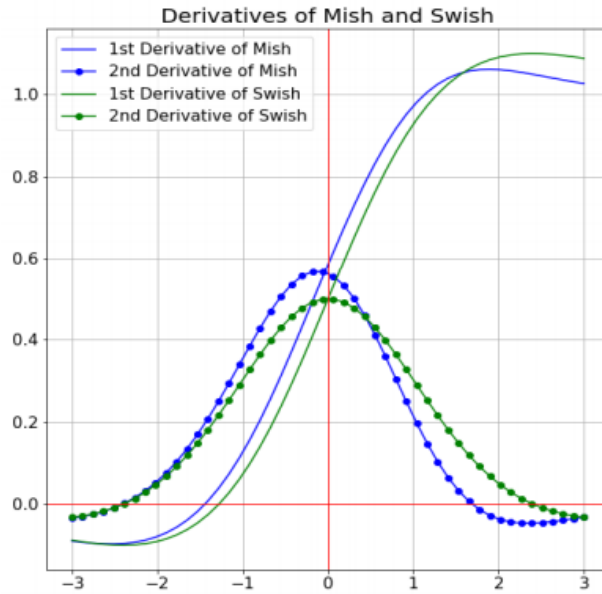
Mish:

$$f(x) = x * \tanh(\text{softplus}(x))$$

$$\text{softplus}(x) = \ln(1 + e^x)$$



دقت آن در عمل از Swish , sigmoid , Relu بهتر است.

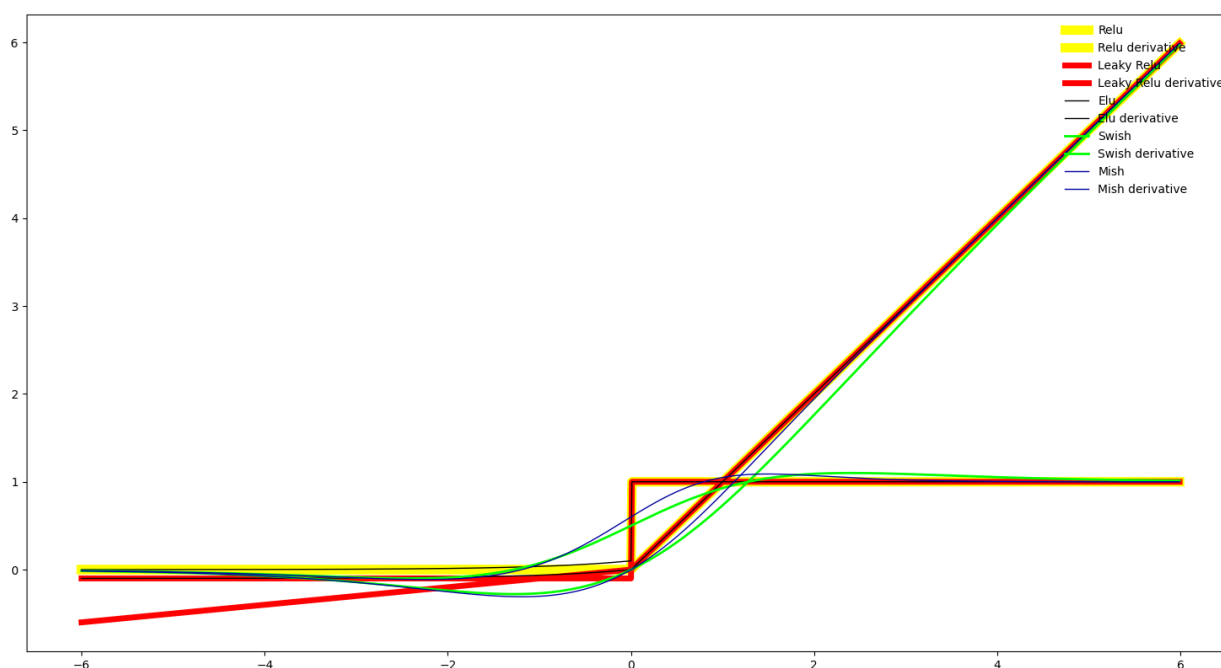


(پ)

$$\begin{aligned}
 \text{ReLU} &\rightarrow R(x) = \max(0, x) \rightarrow \frac{\partial R}{\partial x} = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases} \\
 \text{Leaky ReLU} &\rightarrow R(x) = \max(kx, x) \rightarrow \frac{\partial R}{\partial x} = \begin{cases} 1 & x > 0 \\ k & x < 0 \end{cases} \\
 \text{ELU} &\rightarrow R(x) = \begin{cases} x & x > 0 \\ \alpha(e^x - 1) & x < 0 \end{cases} \rightarrow \frac{\partial R}{\partial x} = \begin{cases} 1 & x > 0 \\ \alpha e^x & x < 0 \end{cases} \\
 \text{Mish} &\rightarrow R(x) = x * \tanh(\ln(1 + e^x)) \rightarrow \frac{\partial R}{\partial x} = \tanh(\ln(1 + e^x)) + x \operatorname{sech}^2(\tanh(\ln(1 + e^x))) \frac{e^x}{1 + e^x} \\
 \text{Swish} &= x * \operatorname{sigmoid}(x) = R(x) \rightarrow \frac{\partial R}{\partial x} = \operatorname{sigmoid}(x) + x * \operatorname{sigmoid}(x) (1 - \operatorname{sigmoid}(x))
 \end{aligned}$$

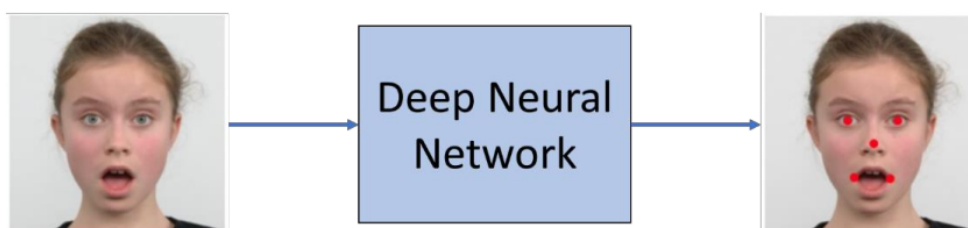
(ت)

کد کامل attach شده.



سوال دوم:

۲. یافتن مکان نقاط مهم در چهره یکی از مراحل بسیار مهم در الگوریتم‌های تحلیل چهره است. همانطور که در شکل زیر نشان داده شده است، ورودی چنین شبکه‌ای یک تصویر برش خورده از چهره است و خروجی آن تخمینی از مختصات نقاط مورد نظر است. در این مثال، خروجی مختصات ۵ نقطه شامل مرکز دو چشم، مرکز بینی و گوشه‌های دهان بوده است که برای نمایش بهتر، بر روی تصاویر اصلی رسم شده‌اند (خروجی شبکه یک تصویر نیست، بلکه خروجی مختصات نقاط است و برای نمایش بهتر بر روی تصویر رسم شده‌اند).



الف) اگر بخواهیم چنین شبکه‌ای طراحی کنیم، در لایه آخر شبکه چند نورون باید داشته باشیم؟ به نظر شما بهتر است از چه تابع فعال‌سازی در لایه آخر استفاده کنیم؟ تابع ضرر مناسب برای حل این مسئله به نظر شما چیست؟ لطفا پاسخ‌های خود را به جزئیات توضیح دهید.

ب) این کد را بررسی کنید و مشخص کنید در آن برای حل مسئله بالا از چه تابع فعال‌سازی و از چه تابع ضرری استفاده کرده است.

ابف) در لایه آخر ۱۰ نورون حداقل نیاز داریم تا بتوانند مختصات ۵ نقطه را بدهند (x, y) به عنوان تابع فعال سازی بهتر است از تابع **linear** استفاده کنیم چون ما نیاز داریم خروجی ما هر عددی بتواند باشد پس **sigmoid** یا **softmax** یا **tanh** که بین یک محدوده محاصره اند توابع خوبی نیستند. تابع **Relu** هم با **activation** خطی **linear** هم گزینه ی بدی نیست.

تابع خطا به نظر **mse** یا **mae** گزینه ی خوبی هستند چون تقریباً با چند **train** نزدیک مکان های مهم میشویم و مفهوم مشابهی دارند اما **cross entropy** گزینه ی خوبی نیست چون اگر خطا کنیم گرادیان به شدت سنگین است و ما را از هدف و نقطه بهینه به شدت دور میکند.

(ب)

برای تعیین ۷۶ ، **landmark points** تعداد 152 نورون در لایه خروجی دارد با تابع فعالسازی **sigmoid**. و تابع ضرر مورد استفاده فاصله ی اقلیدسی بین مقدار صحیح و مقدار پیشبینی شده در هر **mini-batch**

سوال سوم:

۳. این **کد** برای دسته بندی تصاویر مربوط به دو کلاس گربه و سگ نوشته شده است. در این کد تنها نیاز است تا تعداد نورون های لایه آخر، تابع فعال سازی لایه آخر و تابع ضرر را مشخص کنید و کد را اجرا کنید تا نتیجه دسته بندی بدست بیاید. لطفاً حالت های مختلف را ارزیابی کنید و نتایج را مقایسه و تحلیل کنید (می توان مسئله را با ۱ یا ۲ نورون مدل کرد و از توابع فعال سازی مختلفی استفاده کرد).

با توجه به این که مسئله دو کلاسی است یک راه حل خوب میتواند استفاده از **sigmoid** , **binary-crossentropy** میباشد.

چون این دو مکمل هم هستند و هم را جبران میکنند و گرادیان به نسبت خوبی دارند.

راه حلی دیگری که شاید منطقی باشد استفاده از **mse lost function** , **linear activation** می باشد.

در کل به علت طول کشیدن بیش از اندازه زمان ران همه ی موارد تست نشد ولی احتمالاً بهترین ها همین ها هستند.

فایل شبیه سازی در **tensorboard** ضمیمه شده است.

سوال چهارم:

۴. مجموعه داده [Mobile Price Classification](#) را در نظر بگیرید. هدف از این مجموعه داده، تخمین رنج قیمت موبایل با توجه به مشخصات آن مشخص شود (در صورت لزوم اطلاعات بیشتر از این [لینک](#) استفاده کنید). قسمت مشخص شده در این [کد](#) را تکمیل کنید (با انتخاب تعداد نوروں، تابع فعال سازی و تابع ضرر مناسب) و نتایج را گزارش کنید.

این مثال خیلی شبیه مثال تخمین قیمت خرید خانه است که از نوع رگرسیون و پیوسته است.

اولاً دقت کنید پارامترها واحد های اندازه گیری متفاوتی دارند پس در ابتدا ما آن ها را نرمال سازی میکنیم.

برای نوع مسایل رگرسیون بهتر است از تابع خطای mse ، تابع فعال سازی $linear$ یا $relu$ استفاده کرد. و نوروں خروجی تنها یک مقدار باشد که آن هم قیمت تخمین زده شده می باشد.