Homework part 2 - 99222084

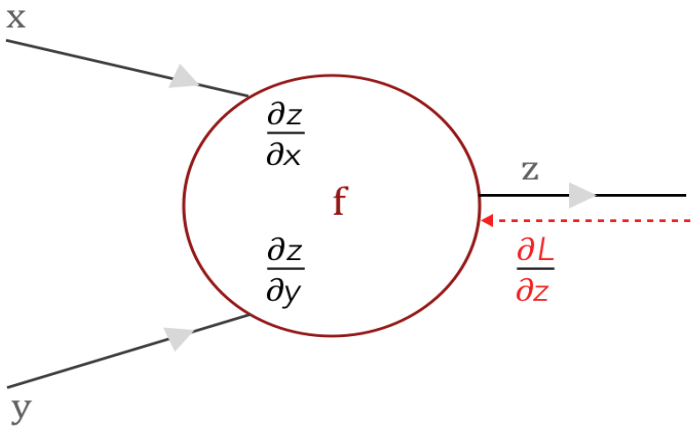
1-Describe the backpropagation details in the convolutional layers.

Simple explanation of backpropagation:

در ابتدا بهتر است بدانیم بک پروپگیشن چگونه کار میکند. بک پروپگیشن یا انتشار رو به عقب خطاها یا به اختصار پس انتشار الگوریتمی برای یادگیری نظارتی شبکه عصبی با استفاده از گرادیان کاهشی میباشد. در واقع از این موضوع می‌آید که محاسبه گرادیان به صورت رو به عقب در شبکه انجام می‌شود و گرادیان لایه خروجی وزن‌ها در ابتدا و گرادیان گرادیان یک لایه برای گرادیان لایه قبلی لایه ورودی در آخر انجام می‌شود؛ بدین صورت که از محاسبات متشقات جزئی استفاده می‌شود. این حرکت رو به عقب اطلاعات خطا، منجر به محاسبه کارآمد گرادیان در هر لایه نسبت به حالتی می‌شود که در آن گرادیان لایه‌ها به صورت جداگانه به دست می‌آید.

Chain rule in convolutional layer:

گراف زیر را در نظر بگیرید:



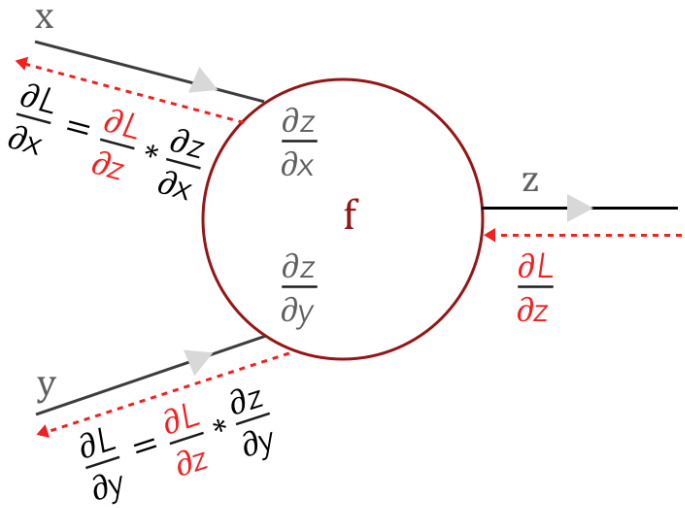
$\frac{\partial z}{\partial x}$ & $\frac{\partial z}{\partial y}$ *are local gradients*

$\frac{\partial L}{\partial z}$ *is the loss from the previous layer which has to be backpropagated to other layers*

در اینجا ما برای فوروارد در سراسر سی ان ان حرکت میکنیم و در پایان با استفاده از تابع ضرر، ضرر را بدست میاوریم.

هنگامی که شروع به بدست اوردن ضرر در بک وارد میکنیم، گرادیان لاس را از لایه قبلی به صورت $\partial L/\partial z$ کم میکنم. سپس برای اینکه ضرر به گیت های دیگر هم منتشر شود باید $\partial L/\partial x$ و $\partial L/\partial y$ را بدست اوریم.

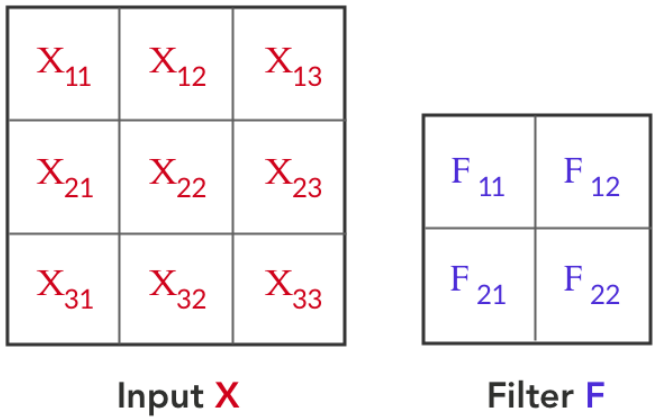
قانون زنجیره ای برای محسابه این دو به کمک می اید.



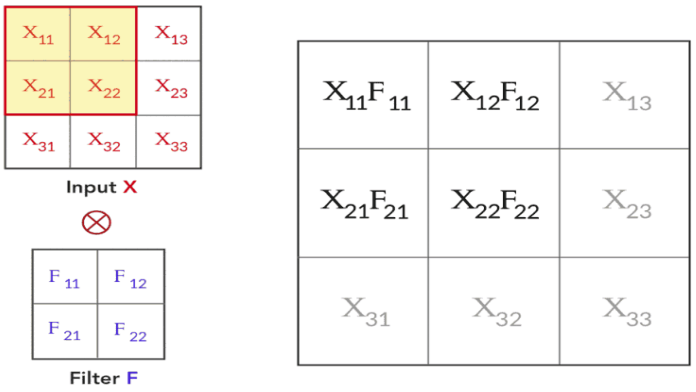
$\frac{\partial z}{\partial x}$ & $\frac{\partial z}{\partial y}$ *are local gradients*

$\frac{\partial L}{\partial z}$ *is the loss from the previous layer which has to be backpropagated to other layers*

حال فرض کنیم همین فانکشن یک فانکشن کانولوشنال و اف فیلتر ما شود که به صورت زیر در نظر میگیریم:

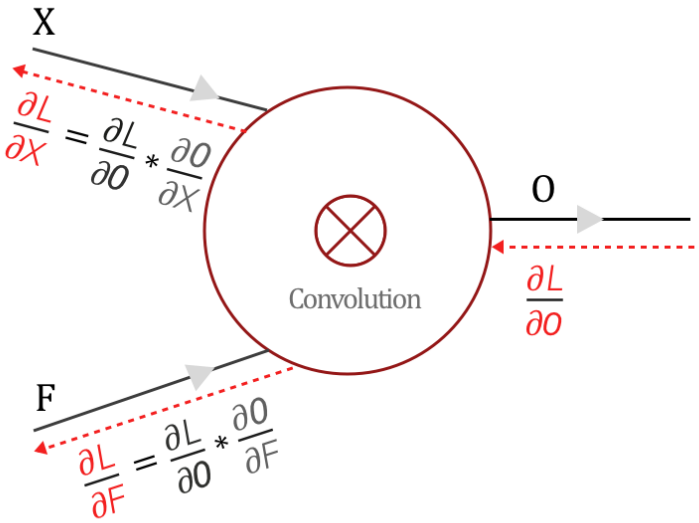


که در نهایت خروجی نهایی فانکشن کانولوشنال ما خواهد شد:



$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$

که این فوروارد پس را به ما میدهد. همان طور که گفته شد ما گرادیان لاس ها را با توجه به خروجی از لایه های قبلی به لایه بعدی با استفاده از $\partial L/\partial O$ با استفاده از گذر به عقب بدست میاوریم پس برای بک وارد پس خواهیم داشت:



$\frac{\partial O}{\partial X}$ & $\frac{\partial O}{\partial F}$ *are local gradients*

$\frac{\partial L}{\partial z}$ *is the loss from the previous layer which has to be backpropagated to other layers*

حال میتوانیم $\partial L/\partial x$ و $\partial L/\partial F$ را بدست اوریم:

در مرحله اول $\partial O/\partial F$ را بدست میاوریم. این به این معناست که باید خروجی را از فیلتر اف کم کنیم. پس یکی یکی خروجی هارا از فیلتر خودشان کم میکنیم:

Local Gradients ⟶ (A)

$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$

Finding derivatives with respect to F_{11} , F_{12} , F_{21} *and* F_{22}

$$\frac{\partial O_{11}}{\partial F_{11}} = X_{11} \quad \frac{\partial O_{11}}{\partial F_{12}} = X_{12} \quad \frac{\partial O_{11}}{\partial F_{21}} = X_{21} \quad \frac{\partial O_{11}}{\partial F_{22}} = X_{22}$$

Similarly, we can find the local gradients for O_{12} , O_{21} *and* O_{22}

در مرحله دوم با استفاده از قانون زنجیره ای خواهیم داشت:

For every element of F

$$\frac{\partial L}{\partial F_i} = \sum_{k=1}^{M} \frac{\partial L}{\partial O_k} * \frac{\partial O_k}{\partial F_i}$$

که اگر انرا باز کنیم خواهد شد:

$$\frac{\partial L}{\partial F_{11}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{11}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{11}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{11}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{11}}$$

$$\frac{\partial L}{\partial F_{12}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{12}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{12}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{12}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{12}}$$

$$\frac{\partial L}{\partial F_{21}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{21}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{21}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{21}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{21}}$$

$$\frac{\partial L}{\partial F_{22}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{22}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{22}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{22}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{22}}$$

و با جایگذاری مقادیر خواهیم داشت:

$$\frac{\partial L}{\partial F_{11}} = \frac{\partial L}{\partial O_{11}} * X_{11} + \frac{\partial L}{\partial O_{12}} * X_{12} + \frac{\partial L}{\partial O_{21}} * X_{21} + \frac{\partial L}{\partial O_{22}} * X_{22}$$

$$\frac{\partial L}{\partial F_{12}} = \frac{\partial L}{\partial O_{11}} * X_{12} + \frac{\partial L}{\partial O_{12}} * X_{13} + \frac{\partial L}{\partial O_{21}} * X_{22} + \frac{\partial L}{\partial O_{22}} * X_{23}$$

$$\frac{\partial L}{\partial F_{21}} = \frac{\partial L}{\partial O_{11}} * X_{21} + \frac{\partial L}{\partial O_{12}} * X_{22} + \frac{\partial L}{\partial O_{21}} * X_{31} + \frac{\partial L}{\partial O_{22}} * X_{32}$$

$$\frac{\partial L}{\partial F_{22}} = \frac{\partial L}{\partial O_{11}} * X_{22} + \frac{\partial L}{\partial O_{12}} * X_{23} + \frac{\partial L}{\partial O_{21}} * X_{32} + \frac{\partial L}{\partial O_{22}} * X_{33}$$

اگر با دقت نگاه کنیم متوجه میشیم $\partial L/\partial F$ چیزی به جز کانولوشن بین ورودی و گرادیان لاس از لایه قبلی نمیشود.

برای بدست اوردن $\partial L/\partial x$ ابتدا $\partial O/\partial x$ را بدست می اوریم.

Local Gradients: → B

$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$

Differentiating with respect to X_{11}, X_{12}, X_{21} *and* X_{22}

$$\frac{\partial O_{11}}{\partial X_{11}} = F_{11} \quad \frac{\partial O_{11}}{\partial X_{12}} = F_{12} \quad \frac{\partial O_{11}}{\partial X_{21}} = F_{21} \quad \frac{\partial O_{11}}{\partial X_{22}} = F_{22}$$

Similarly, we can find local gradients for O_{12}, O_{21} *and* O_{22}

با استفاده از قانون زنجیره ای خواهیم داشت:

For every element of X_i

$$\frac{\partial L}{\partial X_i} = \sum_{k=1}^{M} \frac{\partial L}{\partial O_k} * \frac{\partial O_k}{\partial X_i}$$

که اگر انرا باز کنیم میشود:

$$\frac{\partial L}{\partial X_{11}} = \frac{\partial L}{\partial O_{11}} * F_{11}$$

$$\frac{\partial L}{\partial X_{12}} = \frac{\partial L}{\partial O_{11}} * F_{12} + \frac{\partial L}{\partial O_{12}} * F_{11}$$

$$\frac{\partial L}{\partial X_{13}} = \frac{\partial L}{\partial O_{12}} * F_{12}$$

$$\frac{\partial L}{\partial X_{21}} = \frac{\partial L}{\partial O_{11}} * F_{21} + \frac{\partial L}{\partial O_{21}} * F_{11}$$

$$\frac{\partial L}{\partial X_{22}} = \frac{\partial L}{\partial O_{11}} * F_{22} + \frac{\partial L}{\partial O_{12}} * F_{21} + \frac{\partial L}{\partial O_{21}} * F_{12} + \frac{\partial L}{\partial O_{22}} * F_{11}$$

$$\frac{\partial L}{\partial X_{23}} = \frac{\partial L}{\partial O_{12}} * F_{22} + \frac{\partial L}{\partial O_{22}} * F_{12}$$

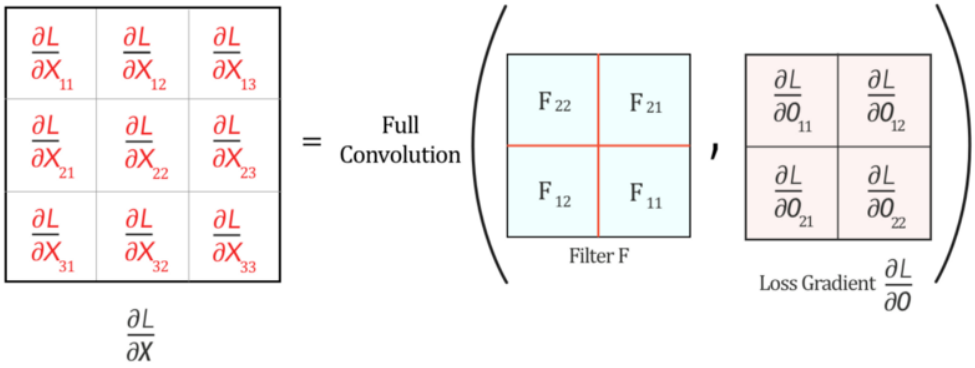
$$\frac{\partial L}{\partial X_{31}} = \frac{\partial L}{\partial O_{21}} * F_{21}$$

$$\frac{\partial L}{\partial X_{32}} = \frac{\partial L}{\partial O_{21}} * F_{22} + \frac{\partial L}{\partial O_{22}} * F_{21}$$

$$\frac{\partial L}{\partial X_{33}} = \frac{\partial L}{\partial O_{22}} * F_{22}$$

که در نهایت ما $\partial L/\partial x$ را خواهیم داشت. این عملیات همان عملیات کانولوشن خواهد بود.

از انجایی که فول کانولوشن $\partial L/\partial x$ را تولید میکند خواهیم داشت:



حال ما هم $\partial L/\partial F$ و هم $\partial L/\partial x$ را داریم. نتیجه نهایی خواهد شد که هم بک وارد و هم فوروارد کانولوشن خواهند بود:

Backpropagation in a Convolutional Layer of a CNN

Finding the gradients:

$$\frac{\partial L}{\partial F} = \text{Convolution}\left(\text{Input } X\,,\ \text{Loss gradient } \frac{\partial L}{\partial O}\right)$$

$$\frac{\partial L}{\partial X} = \text{Full Convolution}\left(180^{\circ}\text{rotated Filter } F\,,\ \text{Loss Gradient } \frac{\partial L}{\partial O}\right)$$

What is the symmetry breaking phenomenon in artificial neural networks? What can be done to prevent this from happening?

در شبکه عصبی مدل هایی که طراحی می کنیم باید به گونه ای کار کنند که ابتدا داده ها را به عنوان ورودی دریافت کنند و در نهایت پردازش آنها در لایه های پنهان خروجی را نمایش دهند و در نهایت با بک وارد بتوانند وزن ها را اپدیت کند تا به یک مدل ایده ال تبدیل شود.

مقدار دهی اولیه وزن ها تأثیر بسیار مهمی بر روند یادگیری و بهبود مدل خواهد داشت. ساده ترین راه برای مقداردهی اولیه وزن ها این است که همه آنها با بایوس به صفر مقداردهی شود. شروع وزن ها و بایاس ها به صفر مطمئناً ما را به مشکل نورون مرده سوق می دهد.

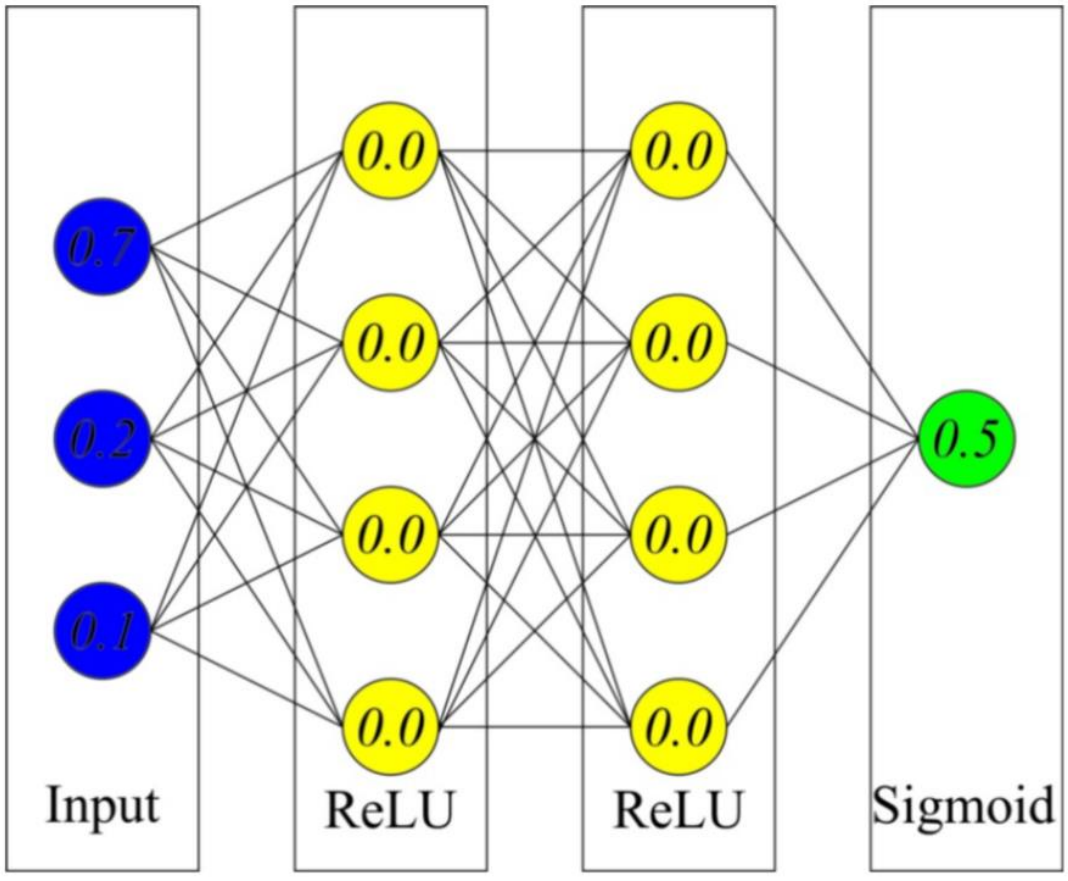


Image by author

برای حل این مشکل می توان مقدار بایوس هر نورون را برابر یک قرار داد. وزن ها قطعا تغییر خواهد کرد. زیرا نورون های رلو خروجی غیر صفر تولید می کنند، اما تغییرات ایجاد شده نامناسب خواهد بود. از سوی دیگر، هر نورون در همان لایه رفتار و وزن یکسانی خواهد داشت. این پدیده را مسئله متقارن می نامند.

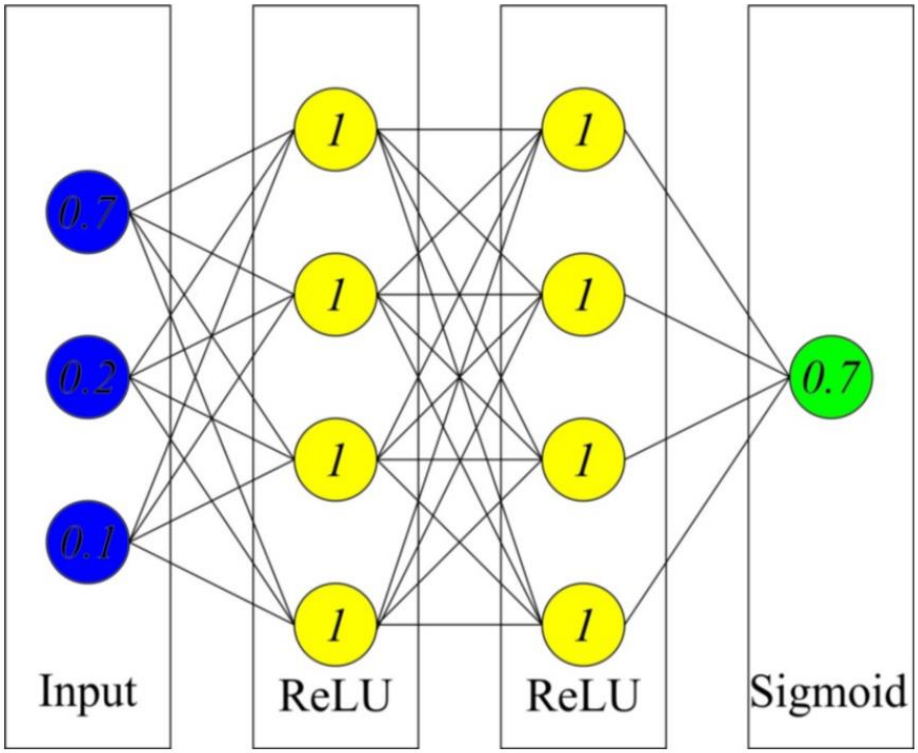
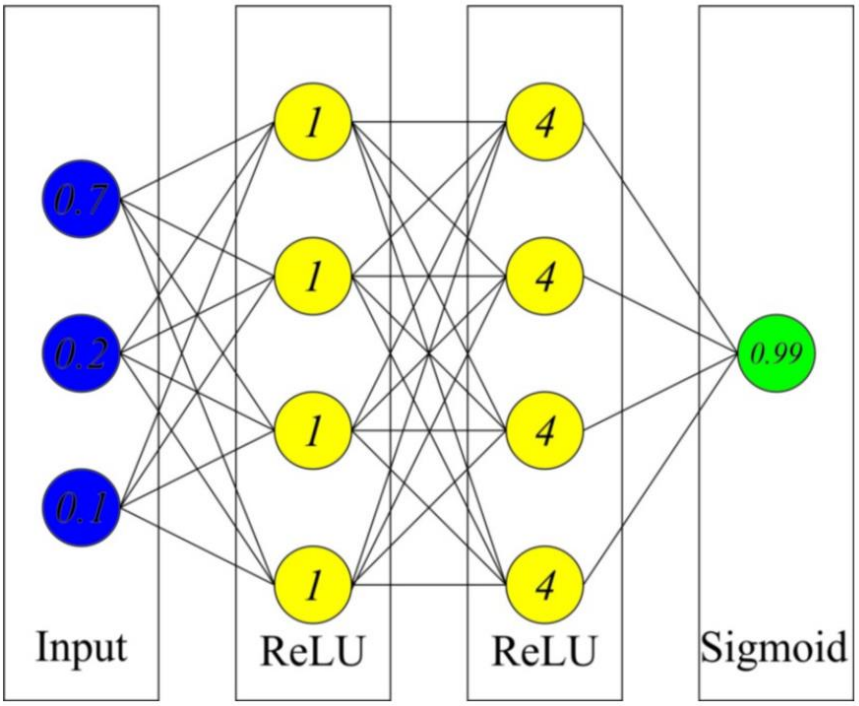
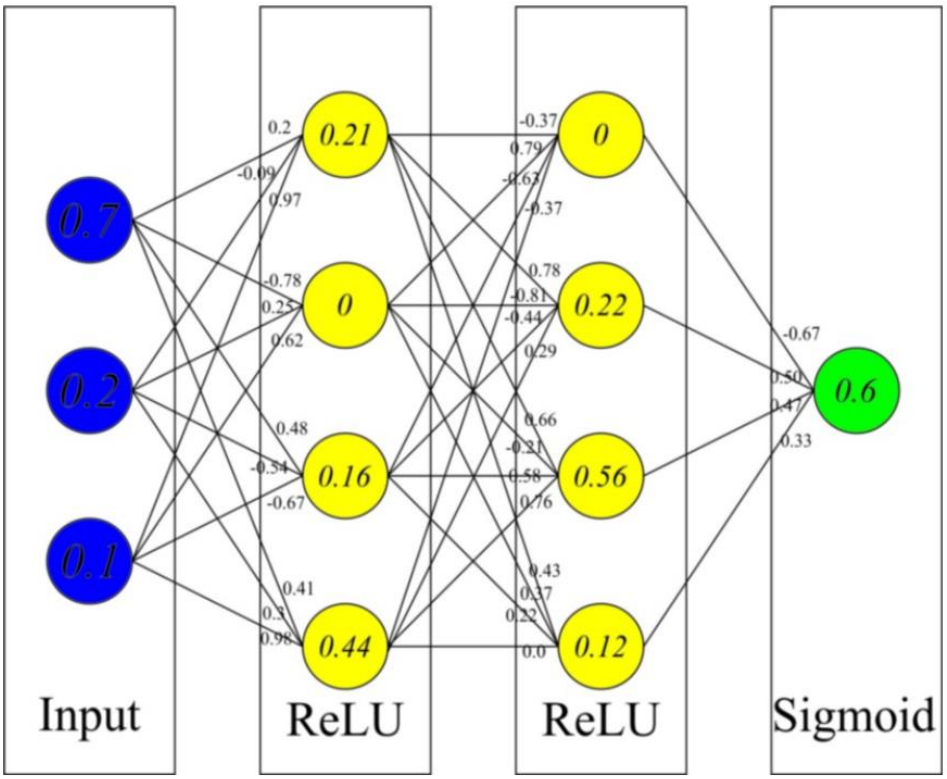


Image by author

این مشکل حتی با مقداردهی اولیه برای وزن ها نیز حل نمی شود.



اگر همه وزن‌ها یکسان شوند هم چیز بدی است زیرا به این معنی است که هر نورون در یک لایه نشان‌دهنده یک ویژگی است. بنابراین، افزودن نورون‌های بیشتر در یک لایه، پیچیدگی شبکه عصبی ما را افزایش نمی‌دهد، زیرا چنین لایه‌ای که گویی فقط ۱ نورون دارد. راه حل برای رفع این مشکل بسیار ساده است، فقط وزن اولیه را تصادفی کنید. برای مثال وزن را تصادفی می کنیم و بایاس را صفر می کنیم.



مقداردهی اولیه تصادفی به ما این امکان را می دهد که این تقارن را بشکنیم. این همچنین به ما اجازه می دهد تا کاری کنیم که همه نورون های شبکه عصبی رفتار متفاوتی داشته باشند.

Random normal:

برای داشتن یک محدوده خاص، می توانیم وزن را با توزیع نرمال تصادفی مقداردهی اولیه کنیم. به جای اینکه وزن را فقط با یک عدد تصادفی مقداردهی اولیه کنیم، با استفاده از توزیع نرمال، می‌توانیم با تنظیم میانگین و انحراف معیار، وزن را با یک محدوده خاص مقداردهی کنیم. به عنوان مثال، با قرار دادن میانگین بر روی صفر و انحراف استاندارد یک عددی در حدود 1- و 1 به ما می دهد.

Random normal:

یک تکنیک بهتر برای مقداردهی اولیه یک شبکه عصبی، کنترل واریانس خروجی است. ما می خواهیم خروجی لایه ای که توسط نورون ها تولید می شود از همان توزیع پیروی کند. مقداردهی اولیه خاویر تکنیکی است که وزن ها را به گونه ای مقداردهی می کند که خروجی تولید شده توسط نورون ها همگی از توزیع یکسانی پیروی کند.

What are the benefits of the pooling layers? What are the drawbacks of the pooling layers? Are you willing to use these layers? Can you use these layers frequently?

لایه های ادغام برای کاهش ابعاد فیچر مپ استفاده می شود. بنابراین، تعداد پارامترهای یادگیری و میزان محاسبات انجام شده در شبکه را کاهش می دهد. لایه ادغام ویژگی های موجود در یک منطقه از فیچر مپ ایجاد شده توسط یک لایه کانولوشن را خلاصه می کند. بنابراین، عملیات بیشتر بر روی ویژگی‌های خلاصه شده به جای ویژگی‌های دقیقاً موقعیت‌یافته تولید شده توسط لایه کانولوشن انجام می‌شود. این باعث می شود که مدل نسبت به تغییرات در موقعیت ویژگی ها در تصویر ورودی قوی تر باشد.

اگر چه یک لایه مکس پولینگ به افزایش تغییرپذیری در تغییر موقعیت و شرایط نور کمک می کند، اما وضوح نقشه ویژگی را کاهش می دهد و اگر بیشتر پیکسل های یک منطقه ادغام بزرگی بالایی داشته باشند، می تواند باعث از بین رفتن اطلاعات در مورد ویژگی های متمایز شود.

استفاده از پولینگ به شرط عدم استفاده مکرر می تواند بسیار مفید و کاربردی باشد زیرا همانطور که گفته شد ممکن است باعث از بین رفتن اطلاعات حیاتی شود. از طرفی پولینگ جزئیات کلی عکس را که مرکز عکس است در مکس پولینگ در ابعاد و پارامترهای کوچکتر نشان می دهد بنابراین ممکن است تکرار آن توجه و اهمیت را از قسمت اصلی بگیرد. استفاده از پدینگ مناسب ممکن است موثر نباشد، بنابراین باید در زمینه های مختلف و در مکان مناسب با دانستن کاربرد و اصول هر پولینگ استفاده شود. حتی مدل های بزرگ امروزه از پولینگ استفاده می کنند. استفاده صحیح از اینها علاوه بر دانش به تجربه نیز بستگی دارد.

Please describe the differences between the cross-entropy vs. quadratic cost. Which one do you prefer for the classification problem?

چگونگی تغییر سرعت یادگیری در این دو متفاوت میشود. به طور خاص، زمانی که ما از کوادریتیک کاست استفاده می کنیم، زمانی که نورون به طور واضح اشتباه می کند، یادگیری آهسته تر از بعد از ان خواهد شد، زیرا نورون به خروجی صحیح نزدیک می شود. در حالی که با کراس انتروپی، هنگامی که نورون به طور واضح اشتباه می کند، یادگیری سریعتر است.

کراس انتروپی لاس در کارهای کلسیفیکیشن استفاده می‌شود که در آن سعی می‌کنیم با به حداکثر رساندن مقدار مورد انتظار برخی تابع در داده‌های ترینینگ، احتمال یک کلاس منفی را به حداقل برسانیم. ایده این تابع ضرر، دادن جریمه بالا برای پیش‌بینی‌های اشتباه و جریمه کم برای طبقه‌بندی صحیح است.

There is an alternative activation function for ReLU called leaky ReLU. I would like you to compare them.

· Which one is faster?

از نظر ظاهری تفاوت چندانی ندارد اما مدل لیکی رلو در ترین با ایپاک های بالاتر تاثیرات خود را نشان می دهد زیرا مانند رلو تراکم داده را در صفر افزایش نمی دهد و با شیب بسیار کوچک و معقول به سمت صفر میل می کند، می تواند عمل کند. در ترین سرعتی بهتر است زیرا تعادل بیشتری دارد و ممکن است در تکرارهای بیشتر و بیشتر برخی از گره ها (نورون ها) به نورون های مرده تبدیل شوند تا عملکرد مناسب تری از مدل شکل بگیرد.

· Which one can prevent gradient vanishing, and how?

دو مزیتی که لیکی رلو نسبت به مدل ساده تر خود دارد:

اول، لیکی رلو مسئله مرگ را به خوبی حل می کند

دوم اینکه در ترینینگ سریعتر است.

نزدیک به صفر بودن "میانگین فعال سازی" باعث می شود ترین سریعتر شود. در واقع ناپدید شدن گرادیان ها یعنی وزنه ها به درستی آپدیت نمی شوند و آنقدر کوچک است که نمی توان آن را آموزش داد و مدل چیزی یاد نمی گیرد. چون لیکی رلو همه آنها را برای مقادیر کوچکتر صفر نمی کند و آنها را به یک مقدار کوچک اختصاص می دهد، این مشکل را مانند رلو ایجاد نمی کند.