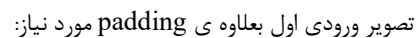


تمرین سری هفتم درس تصویربرداری رقمی

پوریا محمدی نسب

(۴۰۰۷۲۲۱۳۸)

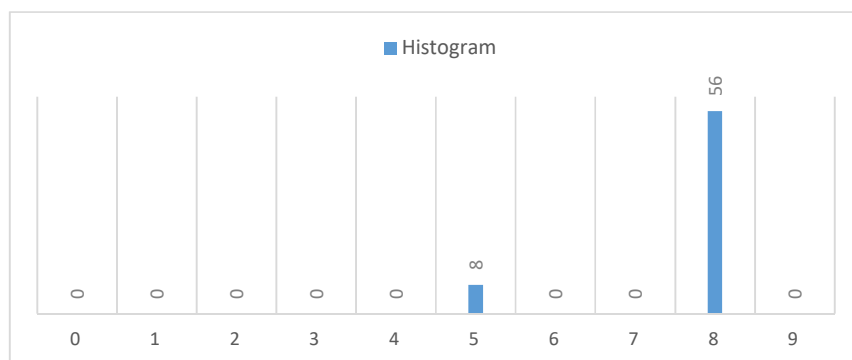
برای کد گذاری از استاندارد زیر استفاده میکنیم:

[illegible]

خروجی تصویر با اعمال LBP :

[illegible]

رسم هیستوگرام LBP:



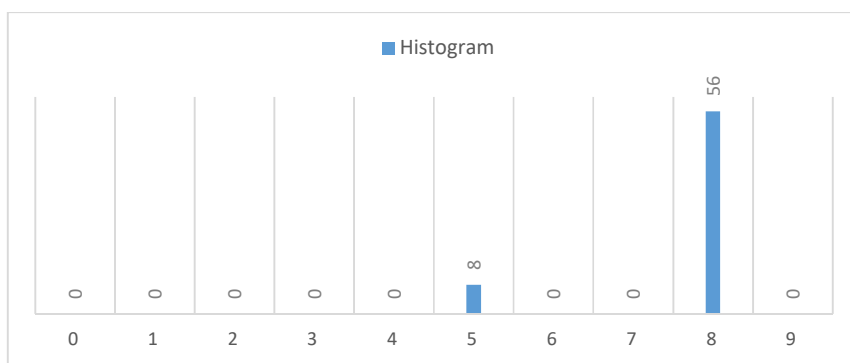
تصویر ورودی دوم بعلاوه ی padding مورد نیاز:

[illegible]

خروجی تصویر با اعمال LBP :

8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
5	5	5	5	5	5	5	5
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8

رسم هیستوگرام LBP:



تحلیل: در این مثال حل شده میتوان به یکی از نقاط ضعف نسخه یکنواخت و مستقل از چرخش LBP اشاره کرد که با وجود لبه در دو جهت مختلف (تصویر اول دارای لبه عمودی و تصویر دوم دارای لبه افقی است). هیستوگرام یکسانی تولید شده است که میتواند در ادامه مشکلاتی برای ماشین یادگیری به عنوان بردار ویژگی ایجاد کند. البته این نکته ی مثبت نیز حائز اهمیت است که حجم کدگذاری بسیار کاهش یافته است.

۲) تابع ضرر Cross entropy یکی از توابع ضرر مناسب برای مسائل دسته‌بندی است. برای آشنایی بیشتر با توابع ضرر و توابع فعالسازی لایه آخر برای حالت‌های مختلف مسائل دسته‌بندی، این لینک را مطالعه کرده و سپس به سوالات زیر پاسخ دهید:

۲- الف) این تابع (Cross Entropy) برای چه نوع مسائل دسته‌بندی ای مناسب است؟

از این تابع دو ورژن مختلف وجود دارد که ورژن باینری آن برای حل مسائل کلاسیک باینری (۲ کلاس) و حالت کلی تر آن برای دسته‌بندی مسائل multi class استفاده میشود.

۲- ب) کمترین مقدار این تابع چه مقداری است؟ این مقدار مربوط به چه حالتی از خروجی شبکه است؟

کمترین مقدار خروجی این تابع 0 است و این مقدار هنگامی رخ میدهد که احتمال کلاس درست 1 تخمین زده شود.

۲- ج) بیشترین مقدار این تابع چه مقداری است (در حالت حدی)؟ این مقدار مربوط به چه حالتی از خروجی شبکه است؟

در حالت حدی اگر احتمال کلاس درست 0 تخمین زده شود، مقدار تابع میتواند بی نهایت شود. این به این معنی است که اگر بازه ای برای خطا نیز تعریف کنیم بیشترین مقدار در آن بازه به عنوان خروجی در این حالت قرار داده میشود.

۲- د) مقدار اولیه این تابع در ابتدای آموزش شبکه چه مقداری است؟ فرض کنید تعداد کلاسها (برچسبهای صحیح) برابر با C است و مقادیر امتیاز این کلاسها نزدیک به یکدیگر هستند.

مقدار اولیه ی این تابع به مقدار دهی اولیه وزن ها و نرمال کردن دیتا وابسته است. به این معنا که به تعداد وزن های رندم، ضرر داریم.

۲- ه) اگر یک شبکه داشته باشیم که لایه آخر آن شامل ۴ نورون خروجی باشد، مقادیر خواسته شده در جدول را محاسبه کنید. \hat{y} پیش بینی شبکه، y برچسب صحیح داده شده و ABCD 4 رقم آخر شماره دانشجویی شما است. برای مثال اگر شماره دانشجویی شما ۴۰۰۷۲۱۲۳۴ باشد؛ $A = 1$ ، $B = 2$ ، $C = 3$ و $D = 4$ هستند.

\hat{y}	$\text{Softmax}(\hat{y})$	y	Cross Entropy Loss
[2,1,3,8]	$\frac{\exp(z_i)}{\sum_j \exp(z_j)}$ [0.0024,0.001,0.0067,0.9899]	[0,0,0,1]	0.0044
		[0,0,1,0]	2.174
		[0,1,0,0]	4
		[1,0,0,0]	2.62

۳) لطفاً به سوالات زیر پاسخ دهید:

۳- الف) تعداد پارامترهای شبکه زیر را محاسبه کنید (محاسبات خود را برای هر لایه به طور دقیق یادداشت کنید).

```
model = Sequential()  
model.add(Input(shape=(500, 7)))  
model.add(Conv1D(filters=16, kernel_size=3, activation="relu"))  
model.add(MaxPool1D())  
model.add(Conv1D(filters=32, kernel_size=5, activation="relu"))  
model.add(MaxPool1D())  
model.add(Conv1D(filters=64, kernel_size=5, activation="relu"))  
model.add(MaxPool1D())  
model.add(Flatten())  
model.add(Dense(units=128, activation="relu"))  
model.add(Dense(units=5, activation="softmax"))
```

Input layer {Param # = 0, Output Shape = (500, 7)}

Conv1D layer {Param # = 352, Output Shape = (498, 16)}

$((\text{previous channels} * \text{kernel size}) + 1) * \text{filters} = ((7 * 3) + 1) * 16 = 352$

MaxPool1D layer {Param # = 0, Output Shape = (249, 16)}

Conv1D layer {Param # = 2592, Output Shape = (245, 32)}

$((\text{previous channels} * \text{kernel size}) + 1) * \text{filters} = ((16 * 5) + 1) * 32 = 2592$

MaxPool1D layer {Param # = 0, Output Shape = (122, 32)}

Conv1D layer {Param # = 10304, Output Shape = (118, 64)}

$((\text{previous channels} * \text{kernel size}) + 1) * \text{filters} = ((32 * 5) + 1) * 64 = 10304$

MaxPool1D layer {Param # = 0, Output Shape = (59, 64)}

Flatten layer {Param # = 0, Output Shape = (3776)}

Dense layer {Param # = 483456, Output Shape = (128)}

$(\text{Number of inputs} * \text{units}) + \text{Units (biases)} = (3776 * 128) + 128 = 483456$

Dense layer {Param # = 645, Output Shape = (5)}

$(\text{Number of inputs} * \text{units}) + \text{Units (biases)} = (128 * 5) + 5 = 645$

Number of all parameters = 497,349

۳ - (ب) لایه کانولوشنال دوبعدی (Conv2D) و سه بعدی (Conv3D) را مقایسه کرده و کاربرد لایه Conv3D را ذکر کنید (امتیازی)

به طور کلی Conv1D برای سیگنال یک بعدی مانند یک سیگنال صوتی استفاده میشود. با این لایه میتوان پترن های مختلف را از درون یک سیگنال یک بعدی استخراج کرد.

Conv2D برای تصاویر استفاده میشود. از دو نوع کانولوشن دیگر معروف تر است. به این لایه ها اصطلاحاً convolution over volume نیز میگویند به این معنی که این لایه ی کانولوشنی روی یک تصویر با چند کانال ورودی (مثلاً RGB) اعمال میشود. در این حالت کرنل نیز باید ۳ بعدی باشد.

Conv3D معمولاً برای فیلم استفاده میشود. در واقع جایی که در هر قطعه از زمان یک فریم تصویر داریم. این نوع لایه ی کانولوشنی از دو نوع دیگر تعداد پارامتر بیشتری برای یادگیری نیاز دارد. دلیل نام گذاری این نوع لایه ها به 3D این است که علاوه بر تمام موارد موجود در یک تصویر، بعد جدید تحت عنوان time نیز خواهیم داشت.

References

- 1) medium.com/@iamvarman/how-to-calculate-the-number-of-parameters-in-the-cnn-5bd55364d7ca
- 2) ccc.illinois.edu/Source_Code/con1d8/con1d-manual-v80.pdf
- 3) programmerall.com/article/6679757991/
- 4) towardsdatascience.com/understanding-and-calculating-the-number-of-parameters-in-convolution-neural-networks-cnns-fc88790d530d
- 5) towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035
- 6) en.wikipedia.org/wiki/Residual_neural_network