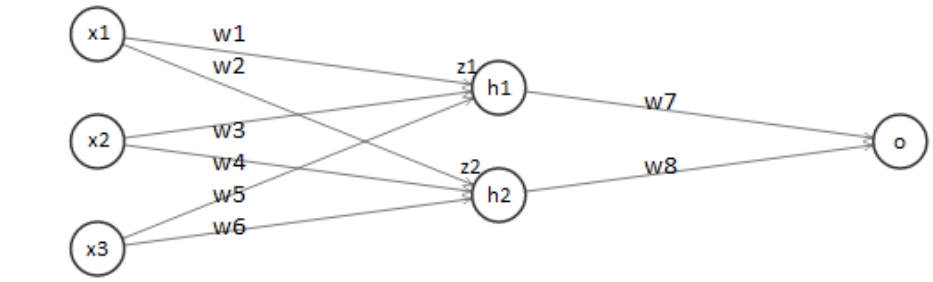
تمارین یادگیری ژرف

سری شماره سه

تمرین ۱

- شبکه عصبی زیر را با وزن های اولیه داده شده درنظر بگیرید و به روش های بهینه سازی "SGD" و "Adam" برای یک تکرار وزن ها را به روش پس انتشار به روز رسانی کنید.



w8	w7	w6	w5	w4	w3	w2	w1
-0.4	0.3	0.4	0.5	0.6	0.7	0.8	0.9

جدول یک. مقداردهی اولیه وزن‌ها

x3	x2	x1	o
1	1	1.5	0

جدول دو. مقداردهی اولیه ورودی‌ها و خروجی

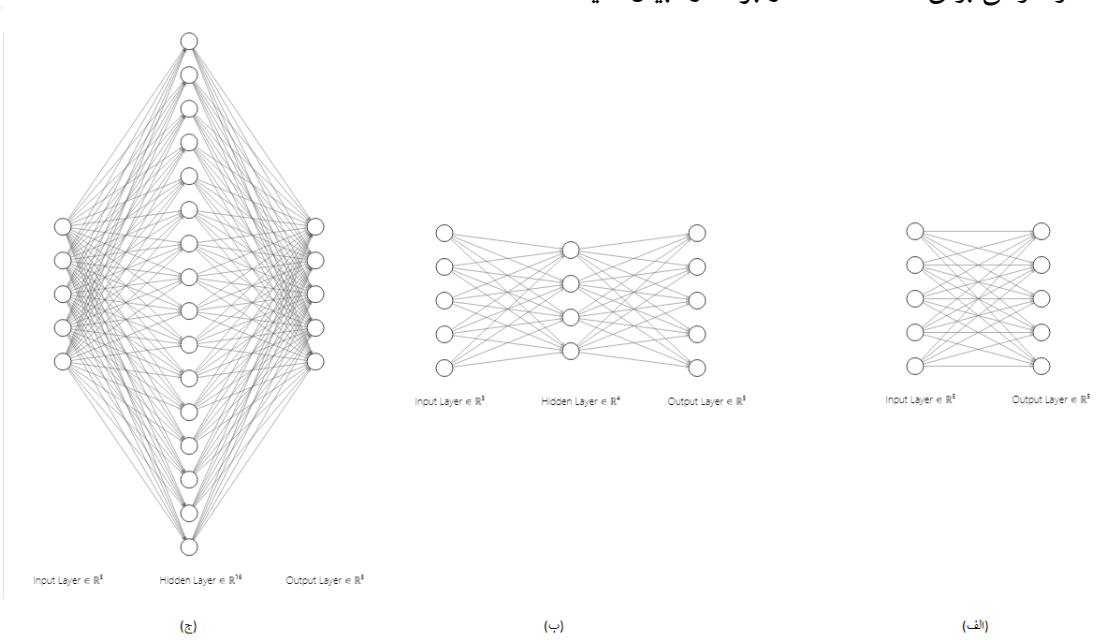
o	h2	h1
Sigmoid	LeakyReLU(0.2)	Tanh

جدول سه. توابع فعالساز مربوط به هر گره

- با طرح یک مثال غیر عددی نشان دهید که پس انتشار چگونه می تواند از مشتق گیری های تکراری پرهیز کند و سرعت بهینه سازی را بالا ببرد.
- بررسی کنید روش "Newton's method" چه مزایایی دارد و نرخ یادگیری (learning rate) دراستفاده از آن به چه صورت است؟ چرا این روش در یادگیری ماشین چندان مورد توجه نیست؟
- مزیت های روش های بهینه سازی "Adam"و "RMSprop" را شرح دهید .

تمرین ۲

در سه شبکه زیر تابع هدف و مجموعه داده آموزشی یکسان هستند و همچنین توابع فعال ساز همگی خطی هستند. تعداد نرون های میانی در مدل «ب»، چهارگره و در شکل «ج» ، شانزده گره می‌باشند. با فرض تعریف وظیفه رگرسیون برای آموزش شبکه ، فرض کنید برای داده (x_i, y_i) پاسخ $\hat{y}_i$ باشد. کیفیت پاسخ های خروجی را قیاس کنید و توجیه ریاضی ارائه دهید. اگر شرطی برای تعداد داده ها وجود دارد بیان کنید.



تمرین ۳

- فرض کنید در یک زیرمسئله ، پس از یک لایه پیچشی با مشخصات زیر در یک مدل یک لایه "AveragePool1D" داریم و و در مدل دیگر یک لایه "MaxPool1D" .به صورت مجزا برای هر یک مدل ها ، ضمن انجام مشتق گیری و بهینه سازی مدل به صورت دستی و عددی ، به صورت کلی نیز توضیح دهید عملیات پس انتشار (Back-Propagation) و به روز رسانی در این لایه ها به چه صورت رخ می‌دهد ؟ (فقط یک مرحله به روز رسانی شود.)

- Modules:
o A: Conv1D (Input-Channel: 2, Output Channel: 1, kernel size: 2, Padding=0, Stride: 1)
o B: Conv1D (Input-Channel: 2, Output Channel: 1, kernel size: 2, Padding=0, Stride: 1)
o C: AveragePool1D (Kernel-Size:2, Padding:0, stride:1)
o D: MaxPool1D (Kernel-Size:2, Padding:0, stride:1)

- Data1:

1 2 -3 2
Channel1

1 2 -2 1
Channel2

- Data2:

4 2 -2 2
Channel1

3 2 -7 2
Channel2

Task$_i$:min || out$_2$ − out_1|| , $out_j = Model_i(Data_j)$ $[forj = 1, 2]$

Optimizer: SGD, learning rate: 1e-4, initialize: Normal distribution with $\mu = 0$ and $\sigma = 0.1$

Model 1: Data -> A -> C -> out
Model 2: Data -> B -> D -> out

- این زیرمسئله را با پیاده سازی به کمک pytorch حل کنید و نزدیکی پاسخ ها را بررسی کنید.

تمرین ۴

الف) شبکه زیر را بر روی مجموعه داده CIFAR10 آموزش دهید. میزان دقت و هزینه برای داده های آموزشی و آزمایشی را به صورت زوج در دو نمودار (یک نمودار برای دقت و یک نمودار برای هزینه) نشان دهید و مقادیر را نیز چاپ کنید همچنین زمان اجرای آموزش را نیز ذخیره کنید و در ادامه موارد بالا را برای پنج آزمایش زیر را در مسئله دخیل کنید.

Epoch-number = 20 | Optimizer : SGD | lr = 0.001 | momentum = 0.9 | batchsize = 32 | loss : Cross Entropy

شبه کد برای مدل :

model: Forward(
Conv2d(in-Channel=3, out-channel=32, kernelSize=3, padding=1),
ReLU,
Conv2d(32, 64, kernelSize=3, stride=1, padding=1),
ReLU,
MaxPool2d(kernelSize=2, stride=2),
BatchNorm2d,

Conv2d(64, 128, kernelSize=3, stride=1, padding=1),
ReLU,
Conv2d(128, 128, kernelSize=3, stride=1, padding=1),
ReLU,
MaxPool2d(kernelSize=2, stride=2),
BatchNorm2d,
Conv2d(128, 256, kernelSize=3, stride=1, padding=1),
ReLU,
Conv2d(256, 256, kernelSize=3, stride=1, padding=1),
ReLU,
MaxPool2d(kernelSize=2, stride=2),
BatchNorm2d,
Flatten,
Linear(? to 1024),
ReLU,
Linear(1024 to 512),
ReLU,
Linear(512 to 10))

۱) منظم ساز نرم یک با ضریب منظم سازی های ۳-e۱ و ۲-e۱

۲) منظم ساز نرم دو با ضریب منظم سازی های ۳-e۱ و ۲-e۱

۳) افزودن داده به روش "DataAugmentation" با کمک عملیات‌های زیر :

۱.چرخاندن ۹۰ درجه ۲. یکی از عملیات های شیفت دادن یا تغییر روشنایی "یا برش تغییر همراه با تغییر اندازه"

torchvision.transforms: RandomResizedCrop/ RandomRotation / ...

۴) افزودن دراپ اوت با نرخ 0.5 به لایه های خطی و با ضریب 0.15 برای سایر لایه ها (تحقیق کنید ، برای یک لایه پیچشی ، چه تفاوتی بین nn.Dropout و nn.Dropout2d وجود دارد؟)

۵)روش Early-stopping

ب) بررسی کنید که افزودن لایه هایی خطی مشابه و پشت سر هم از ۵۱۲ گره به ۵۱۲ گره در محل مناسب و با فعال سازهای برای هر لایه ، می تواند باعث دیده شدن بیش برازش بیشتر در نمودار های خواسته شده برای داده های آموزشی و آزمایشی در محل اول (بدون منظم سازی) شود؟

تمرین ۵

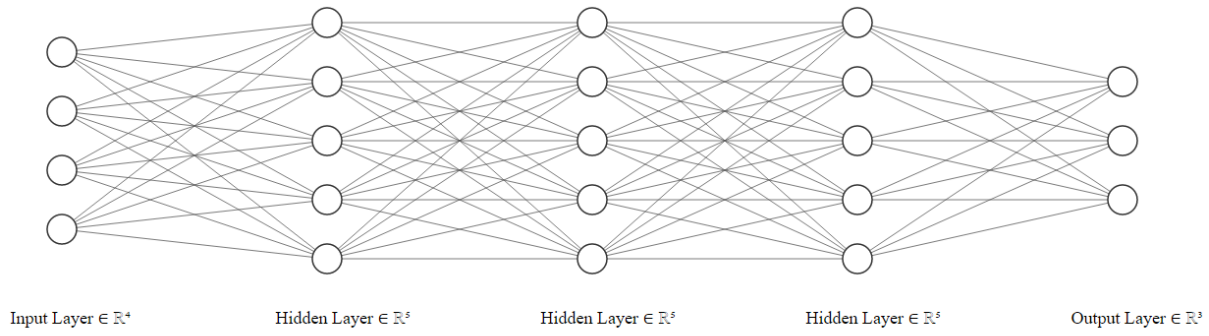
به صورت تئوری نشان دهید که "Dropout " مشابه "Ensemble Methods" عمل می کند.

تمرین ۶

نشان دهید لایه پیچشی زیر مشابه یک MLP روی هر پیکسل عمل می کند :
Conv2D(Input-Channel: M, Output-Channel: N, kernel size: 1, Padding: 0, Stride: 1)

تمرین ۷

- شبکه زیر را بر روی مجموعه داده IRIS آموزش دهید. و خطا و دقت را برای داده های آموزشی وآزمایشی (تست) گزارش کنید.
- سپس یک تابع برای کلاس شبکه بنویسید که مشتقات پس انتشار توسط شما دستی نوشته شده باشد و بدون نیاز به backward و توابع هزینه آماده در پایتورچ (pytodch) با کمک آن تابع درون کلاسی خروجی محاسبه گردد.
- در مورد روش های مقدار دهی اولیه‌ی Xavier initialize و He initialize تحقیق کنید. با فرض جایگزینی فعالساز شبکه با فعالساز sigmoid هر دو روش را اعمال و نتایج را قیاس کنید . همچنین فعال ساز های متناسب با آنها (شهود آزمایشی و عملی) را تحقیق کنید. (بر اساس مقاله [۱])
- آزمایش بخش اول را طوری انجام دهید که در هزارمین لایه عملیات متوقف شود و موارد ضروری جهت آموزش (شامل مقادیر و ماژول های loss و optimizer و model weights) ذخیره گردند و مجددا با فراخوانی موارد ذخیره شده با شروع دوباره برنامه ، آموزش ادامه یابد.



Weight initialize: sample from $UNIFORM\ distribution(low = 0, high = 1)$
Split rate for Test size to Train size = 0.2
Active Functions : Leaky-Relu (0.2) for hidden layers , Softmax for output layer
Loss : Cross Entropy
Optimizer : Adam (lr:0.001)
Epochs : 2000
Nodes in each hidden layer : 5
Base number of hidden layers : 3

تمرین ۸

الف) با مراجعه به مقاله [۲] توضیح دهید که نرمال سازی "Batch Normalization"چه عملی انجام می دهد و چگونه با کاهش "Internal Covariate Shift" باعث سرعت بخشیدن به آموزش مدل می شود؟ چهار مزیت استفاده از این روش را با رجوع به مقاله مرتبط توضیح دهید.
ب) تحقیق کنید (Batch Normalization) باید قبل دراپ اوت (dropout) باشد یا بعد از آن ؟ وضعیت این دو نسبت به فعالساز چگونه است؟ این نتیجه گیری طبق تئوری است یا مشاهده عملی ؟

ب) بررسی کنید "Batch Normalization" در داده های آزمایشی (تست) چگونه مورد استفاده قرار می گیرد و محاسبه می شود.
ج) نشان دهید که یک مدل "Maxout" با فقط دو واحد پنهان می تواند هر تابع دلخواه پیوسته از $v \in R^n$ را تخمین بزند. [۳]
د) توضیح دهید batch normalization در یک لایه پیچشی دو بعدی $tesnor \in \mathbb{R}^{(N,C,H,W)}$چگونه عمل می کند؟
ه) تفاوت های Layer normalization و batch normalization را بررسی کنید.
ی) بررسی کنید که دراپ اوت (dropout) در شبکه CNN به چه صورت مورد استفاده است؟ (در لایه پیچشی)

تمرین ۹

دو شبکه زیر را که در آنها لایه ها تماما خطی با توابع فعالساز غیرخطی هستند ، درنظربگیرید. تفاوت دو شبکه در تابع فعالساز است. تعداد گره های لایه ورودی و خروجی را متناسب با تعداد ویژگی ها و تعداد کلاسهای مربوط به داده و لایه های پنهان توضیح داده شده پیدا کنید و با هدف کلاس بندی، شبکه ها را با تنظیمات و مفروضات مشخص شده آموزش دهید . ضمن گزارش خطا و هزینه برای داده های آموزشی و آزمایشی ، مقدار نهایی (پس از اتمام تکرار ها) نرم یک و نرم دوم گرادیان وزن‌های هر لایه در طول گذار به سمت لایه خروجی در یک نمودار برحسب شماره لایه یادداشت کنید و برای دو شبکه قیاس کنید. این آزمایش برای شبکه های مشابه اما با تعداد لایه های پنهان ۳ و ۹ و ۱۳ و ۱۷ و ۱۹ لایه تکرار شود.

Dataset: Breast Cancer (Use scikit-learn)
Train/test Split: ratio of (number of test samples / number of train samples) = 0.1
Optimizer: Adam, Learning rate=0.001
Loss : Cross-Entropy
Number of epochs: 50
Number of nodes in each hidden layer: 5
Active functions:
for model A: Sigmoid
for model B: ReLU

در قدم بعدی با افرودن (batchnorm) در هر لایه برای هر دو مدل آزمایش را تکرار کنید و نتایج را قیاس کنید.

تمرین ۱۰

الف) شبکه زیر را با خواسته های مشخص شده پیاده سازی کنید و با تغییر میزان اندازه دسته آموزشی (batch size) با اندازه های ۱ و ۸ و ۳۲ و ۵۰، اعوجاج در نمودارهای هزینه - تکرار و خطا - تکرار را برای داده های آموزشی در هر دسته (و در هر تکرار) و برای داده های آزمایشی در حین تکرار های آموزش (در هر تکرار) قیاس کنید. (برای حصول نمودار های مطلوب برای حالت تست ، عملیات مربوط به تست ، باید در حین آموزش در هر تکرار اجرا شود.) (تابع هزینه کراس انتروپی (Cross Entropy)مطلوب است اما با درنظر گرفتن فعالساز لایه خروجی ، باید از تابع دیگری استفاده شود تا برآیند فعالساز خروجی واین تابع خطا معادل تابع هزینه کراس انتروپی تعریف شده در پایتورچ شود.) .
برای تمرین بیشتر می توانید میزان مصرف حافظه (GPU/CPU) و سرعت اجراها را نیز مقایسه کنید.

Data : MNIST (from torchvision[5])
Optimizer:
Epochs : 15
Loss : Cross-Entropy

ب)با کمک torch.optim.lr_scheduler و با درنظر گیری آهنگ مناسب (به عنوان مثال تغییر آهنگ نرخ یادگیری با ضریب gamma=0.9) ، شبکه را با تنظیمات بخش الف اما با batchsize=50 طوری آموزش دهید که نرخ یادگیری در هر تکرار نسبت به تکرار قبلی کاهش یابد و نتایج مطلوب را برای داده های آموزشی و آزمایشی گزارش کنید.

ج) شبکه را با batchsize=32 و با سایر تنظیمات مشابه بخش الف ولی با توابع هزینه زیر آموزش دهید و نتایج را قیاس کنید: (با کمک توابع هزینه از torch.nn)

nn.MultiLabelSoftMarginLoss() ,
nn.MultiMarginLoss() ,
nn.PoissonNLLLoss() ,
nn.GaussianNLLLoss() ,
nn.MultiLabelMarginLoss()

د) فرض کنید که یک مسئله کلاس بندی با دو کلاس دارید. این مسئله را می توان با دو گره خروجی و یا یک گره خروجی پیاده سازی کرد؟ چه تفاوتی وجود دارد؟ درصورت پاسخ مثبت با فرض اینکه هدف از کلاس بندی مجموعه داده بالا صرفا دو کلاسه باشد (یک کلاس برای حضور داده در کلاس اول (صفر بودن رقم دستنویس)و یک کلاس برای عدم حضور در کلاس اول (عدم صفر بودن رقم دستنویس)) با این دو دیدگاه کلاس بندی را انجام دهید و نتایج را مقایسه کنید.

ه)دربخش د ، با توجه به بالانس نبودن تعداد داده ها چه معیار هایی برای ارزیابی های مشابه دقت (Accuracy) برای تحلیل بهتر می توان در نظر گرفت؟ در صورت لزوم در پیاده سازی و نتایج خود این موضوع را درنظر بگیرید.

معماری شبکه :

layer1	Conv2d(in-channels=1, out-channels=16, kernel-size=3, padding=1) , ReLU , MaxPool2d(kernel-size=2,stride=2)
layer2	Conv2d(in-channels=16, out-channels=32, kernel-size=3, padding=1) , ReLU , MaxPool2d(kernel-size=2,stride=2)
-	flatten : view(-1, ?)
layer3	linear(? to 1024), ReLU , dropout
layer4	linear(1024 to 10) , log-softmax

تمرین ۱۱

الف) با کتابخانه "opencv-python" فیلتر های زیر را بر تصویر ضمیمه شده Pic1 اعمال کنید ، نتیجه را نمایش دهید و تحلیل انجام دهید.

1. **Sharpen kernel**

0	-1	0
-1	5	-1
0	-1	0

2. **Laplacian kernel**

0	1	0
1	-4	1
0	1	0

3. **Emboss kernel**

-2	-1	0
-1	1	1
0	1	2

4. **Outline kernel**

-1	-1	-1
-1	8	-1
-1	-1	-1

5. **Bottom sobel**

-1	-2	-1
0	0	0
1	2	1

6. **Right sobel**

-1	0	1
-2	0	2
-1	0	1

7. **Top sobel**

1	2	1
0	0	0
-1	-2	-1

8. **Difference**

-1	-1	-1
-1	8	-1
-1	-1	-1

9. **Weighted average**

1	1	1
1	8	1
1	1	1

ب) تصویر ضمیمه شده Pic2 را به عنوان ماسک درنظربگیرید و با کمک "opencv-python" تطبیق الگو با تصویر Pic1 را به روش های زیر انجام دهید و نتایج را مشاهده کنید.

1) Cross Correlation
2) Normalized Cross Correlation
3) Correlation Coefficient
4) Normalized Correlation Coefficient
5) Square Difference
6) Normalized Square Difference

ج) بررسی کنید که چگونه یک فیلتر پیچشی مشتق گیر تک بعدی [1- ;2; -1] را برای فضای دو بعدی (ماتریسی) باید نوشت.
د) در مورد نحوه کار فیلتر های Canny و bilateralFilter تحقیق کنید و این دو فیلتر را نیز مشابه بخش الف بر تصویر مربوطه اعمال کنید.

*تصاویر از مجموعه داده [۶] انتخاب شده‌اند.

- تمرین ۱۲

جدول زیر مربوط به یک شبکه عصبی شامل لایه های پیچشی و خطی است. با فرض اینکه تعداد کلاس های داده ورودی ۱۰۰۰ مورد است جدول را پر کنید. مدل در حال انجام وظیفه کلاس بندی است.

Layer name (type)	Specification	output dim	number of parameters
INPUT	-	(3,227,227)	-
convolution layer	in=* out=* k=11 p=* s=4	(96,55,55)	*
*	in=* out=* k=3 p=0 s=*	(96,27,27)	0
convolution layer	in=* out=* k=* p=0 s=1	(256,27,27)	*
Max Pooling	k=3 s=2	*	*
convolution layer	in=256 out=384 k=3 p=1 s=1	*	*
convolution layer	in=384 out=* k=3 p=1 s=1	*	1327488
convolution layer	***	(256,13,13)	*
Max Pooling	k=3, ***	(256,6,6)	*
linear	from * to *	-	*
linear	from 4096 to *	-	*
linear	from 4096 to *	*	*

توجه شود به عنوان مثال (3,10,20) به معنی تنسور با ۳ کانال و طول و عرض ۱۰ و ۲۰ می باشد. همچنین p به معنی پدینگ و s به معنی طول گام و k به معنی اندازه فیلتر و in به معنی تعداد کانال های ورودی و out تعداد کانال های خروجی است. اندازه فیلتر پولینگ نیز با k نمایش داده می شود. در شمارش .در اینجا تعداد پارامترهی مربوط به بایاس نیز محسوب شده است به این طریق که برای هر کانال خروجی تنها یک واحد پارامتر بایاس وجود دارد. در صورت عدم شمارش بایاس ، در سطر هفتم تعداد پارامترها به 1327104 میرسد.

تمرین ۱۳

به صورت تئوری ارتباط "Early Stopping" و نرم دو "$l2$-Norm" را بررسی کنید.
(مدل را خطی ساده با بهینه ساز ساده GD و با تابع هزینه درجه دوم "quadratic" فرض کنید.)

تمرین ۱۴

الف) می‌دانیم که یک لایه پیچش دو بعدی به صورت زیر قابل نوشتن است. یک لایه "Separable Convolution" را که شامل دو بخش Depth-wise convolution و Point-wise convolution است ، در قالب مشابه بنویسید.

$$Z(i,:,:) = \sum_{j=1}^{n_k} W(i,j,:,:) * H(j,:,:) + B(i,:,:)$$

- W: Weights of 2d-convolution
H: Previous Layer
Z: Next Layer
B: Bias

ب) وزن معادل یک لایه پیچش را W فرض می کنیم که $W \in \mathbb{R}^{N\times M\times L\times K}$ می باشد. اگر به صورت رنک پایین این تنسور را حاصل ضرب چهار بردار یک بعدی درنظر بگیریم به طوریکه :

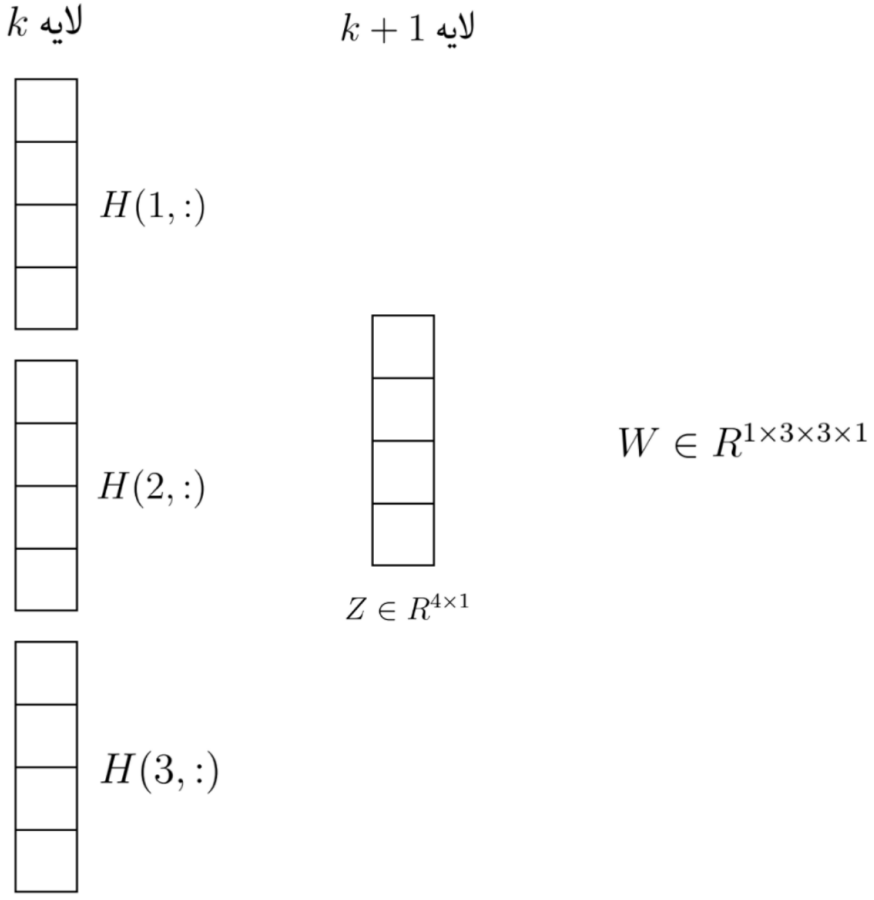
$$W_{i,j,l,k} = a_i b_j c_l d_k$$

آنگاه پیچش را به صورت تئوری در این لایه با کمترین محاسبات بنویسید ، مشتقات مربوط به پس‌انتشار آن را محاسبه کنید و مزیت‌های این مدل‌سازی را شرح دهید.

ج) قطعه کد مربوط به بخش دوم را بنویسید.
د) مشتقات پس‌انتشار برای یک لایه پیچشی ساده را بنویسید.

تمرین ۱۵

در شکل زیر دو لایه k و k+1 از یک شبکه پیچشی یک بعدی را مشاهده می کنید.



ابتدا مشتقات مربوط پس‌انتشار پیچش بین این دو لایه محاسبه کنید و محاسبه مشتق را در این مرحله به صورت یک پیچش مدل کنید. سپس با فرض $Z \in \mathbb{R}^{2\times1}$ باشد نوع پیچش را مشخص کنید و مشتقات مربوط به آنرا محاسبه کنید.

توجه : استفاده از "Pytorch" برای برنامه نویسی شبکه‌های عمیق ضروری است و استفاده از پکیج‌ها یا زبان‌های دیگر ، مشروط به اجازه از استاد درس و هماهنگی با تیم حل تمرین می‌باشد.[۴]

مراجع

[1] He, K., Zhang, X., Ren, S., Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision (pp. 1026-1034).

[2] Ioffe, S., Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). pmlr.

[3] Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y. (2013, May). Maxout networks. In International conference on machine learning (pp. 1319-1327). PMLR.

[4] URL : `https://pytorch.org`

[5] URL : `https://pytorch.org/vision/stable/index.html`

[6] Dataset: "iphone2dslr_flower". This dataset is part of the CycleGAN datasets, originally hosted here:
`https://people.eecs.berkeley.edu/~taesung_park/CycleGAN/datasets/`

[7] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning. MIT press.

دوره یادگیری ژرف / دانشگاه تربیت مدرس / گروه علوم کامپیوتر / نیمسال دوم سال تحصیلی ۱۴۰۲-۱۴۰۱

استاد درس : دکتر منصور رزقی آهق (Mrezghi.ir)

ا ا دستیار آموزشی : سهیل طباطبایی مرتضوی