합성곱 신경망(CNN)

PARA#: 1M pixel= 1M 차원

ㄴhidden pixel: 1000개여도 총 1M*1000=1B -> 학습할게 너무 많음

해결: CNN: 시신경 모방

월도 찾기 게임

patch에 **점수 할당**해 계산

ㄴ 동일 patch에 비슷하게 반응: 이동 불변성

CNN: **합성곱**(Convolutional) layer를 포함

ㄴ1layer 표현: 1B para-> 100 para

img= **높이*너비*channel**

ㄴ Channel: RGB

In [1]:
```
!pip install d2l==1.0.3
```

```
Collecting d2l==1.0.3
  Using cached d2l-1.0.3-py3-none-any.whl.metadata (556 bytes)
Collecting jupyter==1.0.0 (from d2l==1.0.3)
  Using cached jupyter-1.0.0-py2.py3-none-any.whl.metadata (995 bytes)
Collecting numpy==1.23.5 (from d2l==1.0.3)
  Using cached numpy-1.23.5-cp310-cp310-manylinux_2_17_x86_64.manylinux2014
_x86_64.whl.metadata (2.3 kB)
Collecting matplotlib==3.7.2 (from d2l==1.0.3)
  Using cached matplotlib-3.7.2-cp310-cp310-manylinux_2_17_x86_64.manylinux
2014_x86_64.whl.metadata (5.6 kB)
Collecting matplotlib-inline==0.1.6 (from d2l==1.0.3)
  Using cached matplotlib_inline-0.1.6-py3-none-any.whl.metadata (2.8 kB)
Collecting requests==2.31.0 (from d2l==1.0.3)
  Downloading requests-2.31.0-py3-none-any.whl.metadata (4.6 kB)
Collecting pandas==2.0.3 (from d2l==1.0.3)
  Downloading pandas-2.0.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_
x86_64.whl.metadata (18 kB)
Collecting scipy==1.10.1 (from d2l==1.0.3)
  Downloading scipy-1.10.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_
x86_64.whl.metadata (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.9/58.9 kB 2.4 MB/s eta 0:0
0:00
Requirement already satisfied: notebook in /usr/local/lib/python3.10/dist-p
ackages (from jupyter==1.0.0->d2l==1.0.3) (6.5.5)
Collecting qtconsole (from jupyter==1.0.0->d2l==1.0.3)
  Downloading qtconsole-5.6.0-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: jupyter-console in /usr/local/lib/python3.1
0/dist-packages (from jupyter==1.0.0->d2l==1.0.3) (6.1.0)
Requirement already satisfied: nbconvert in /usr/local/lib/python3.10/dist-
packages (from jupyter==1.0.0->d2l==1.0.3) (6.5.4)
Requirement already satisfied: ipykernel in /usr/local/lib/python3.10/dist-
packages (from jupyter==1.0.0->d2l==1.0.3) (5.5.6)
Requirement already satisfied: ipywidgets in /usr/local/lib/python3.10/dist
-packages (from jupyter==1.0.0->d2l==1.0.3) (7.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.1
0/dist-packages (from matplotlib==3.7.2->d2l==1.0.3) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/di
st-packages (from matplotlib==3.7.2->d2l==1.0.3) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.
10/dist-packages (from matplotlib==3.7.2->d2l==1.0.3) (4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.
10/dist-packages (from matplotlib==3.7.2->d2l==1.0.3) (1.4.7)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.1
0/dist-packages (from matplotlib==3.7.2->d2l==1.0.3) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/d
ist-packages (from matplotlib==3.7.2->d2l==1.0.3) (10.4.0)
Collecting pyparsing<3.1,>=2.3.1 (from matplotlib==3.7.2->d2l==1.0.3)
  Downloading pyparsing-3.0.9-py3-none-any.whl.metadata (4.2 kB)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/pytho
n3.10/dist-packages (from matplotlib==3.7.2->d2l==1.0.3) (2.8.2)
Requirement already satisfied: traitlets in /usr/local/lib/python3.10/dist-
packages (from matplotlib-inline==0.1.6->d2l==1.0.3) (5.7.1)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/di
st-packages (from pandas==2.0.3->d2l==1.0.3) (2024.2)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/
dist-packages (from pandas==2.0.3->d2l==1.0.3) (2024.2)
```

```
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/p
ython3.10/dist-packages (from requests==2.31.0->d2l==1.0.3) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/di
st-packages (from requests==2.31.0->d2l==1.0.3) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python
3.10/dist-packages (from requests==2.31.0->d2l==1.0.3) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python
3.10/dist-packages (from requests==2.31.0->d2l==1.0.3) (2024.8.30)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-p
ackages (from python-dateutil>=2.7->matplotlib==3.7.2->d2l==1.0.3) (1.16.0)
Requirement already satisfied: ipython-genutils in /usr/local/lib/python3.1
0/dist-packages (from ipykernel->jupyter==1.0.0->d2l==1.0.3) (0.2.0)
Requirement already satisfied: ipython>=5.0.0 in /usr/local/lib/python3.10/
dist-packages (from ipykernel->jupyter==1.0.0->d2l==1.0.3) (7.34.0)
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.10/
dist-packages (from ipykernel->jupyter==1.0.0->d2l==1.0.3) (6.1.12)
Requirement already satisfied: tornado>=4.2 in /usr/local/lib/python3.10/di
st-packages (from ipykernel->jupyter==1.0.0->d2l==1.0.3) (6.3.3)
Requirement already satisfied: widgetsnbextension~=3.6.0 in /usr/local/lib/
python3.10/dist-packages (from ipywidgets->jupyter==1.0.0->d2l==1.0.3) (3.
6.9)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/
python3.10/dist-packages (from ipywidgets->jupyter==1.0.0->d2l==1.0.3) (3.
0.13)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0
in /usr/local/lib/python3.10/dist-packages (from jupyter-console->jupyter==
1.0.0->d2l==1.0.3) (3.0.48)
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-p
ackages (from jupyter-console->jupyter==1.0.0->d2l==1.0.3) (2.18.0)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packa
ges (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (4.9.4)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/
dist-packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (4.12.3)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-pac
kages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (6.1.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.10/dist
-packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (0.7.1)
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python
3.10/dist-packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (0.4)
Requirement already satisfied: jinja2>=3.0 in /usr/local/lib/python3.10/dis
t-packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (3.1.4)
Requirement already satisfied: jupyter-core>=4.7 in /usr/local/lib/python3.
10/dist-packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (5.7.2)
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python
3.10/dist-packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (0.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.1
0/dist-packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (2.1.5)
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.
10/dist-packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (0.8.4)
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.1
0/dist-packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (0.10.0)
Requirement already satisfied: nbformat>=5.1 in /usr/local/lib/python3.10/d
ist-packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (5.10.4)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/pytho
n3.10/dist-packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (1.5.1)
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.10/dist-p
```

```
ackages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (1.3.0)
Requirement already satisfied: pyzmq<25,>=17 in /usr/local/lib/python3.10/d
ist-packages (from notebook->jupyter==1.0.0->d2l==1.0.3) (24.0.1)
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.10/dis
t-packages (from notebook->jupyter==1.0.0->d2l==1.0.3) (23.1.0)
Requirement already satisfied: nest-asyncio>=1.5 in /usr/local/lib/python3.
10/dist-packages (from notebook->jupyter==1.0.0->d2l==1.0.3) (1.6.0)
Requirement already satisfied: Send2Trash>=1.8.0 in /usr/local/lib/python3.
10/dist-packages (from notebook->jupyter==1.0.0->d2l==1.0.3) (1.8.3)
Requirement already satisfied: terminado>=0.8.3 in /usr/local/lib/python3.1
0/dist-packages (from notebook->jupyter==1.0.0->d2l==1.0.3) (0.18.1)
Requirement already satisfied: prometheus-client in /usr/local/lib/python3.
10/dist-packages (from notebook->jupyter==1.0.0->d2l==1.0.3) (0.21.0)
Requirement already satisfied: nbclassic>=0.4.7 in /usr/local/lib/python3.1
0/dist-packages (from notebook->jupyter==1.0.0->d2l==1.0.3) (1.1.0)
Collecting qtpy>=2.4.0 (from qtconsole->jupyter==1.0.0->d2l==1.0.3)
  Downloading QtPy-2.4.1-py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.1
0/dist-packages (from ipython>=5.0.0->ipykernel->jupyter==1.0.0->d2l==1.0.
3) (71.0.4)
Collecting jedi>=0.16 (from ipython>=5.0.0->ipykernel->jupyter==1.0.0->d2l=
=1.0.3)
  Using cached jedi-0.19.1-py2.py3-none-any.whl.metadata (22 kB)
Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-
packages (from ipython>=5.0.0->ipykernel->jupyter==1.0.0->d2l==1.0.3) (4.4.
2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dis
t-packages (from ipython>=5.0.0->ipykernel->jupyter==1.0.0->d2l==1.0.3) (0.
7.5)
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-p
ackages (from ipython>=5.0.0->ipykernel->jupyter==1.0.0->d2l==1.0.3) (0.2.
0)
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dis
t-packages (from ipython>=5.0.0->ipykernel->jupyter==1.0.0->d2l==1.0.3) (4.
9.0)
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.
10/dist-packages (from jupyter-core>=4.7->nbconvert->jupyter==1.0.0->d2l==
1.0.3) (4.3.6)
Requirement already satisfied: notebook-shim>=0.2.3 in /usr/local/lib/pytho
n3.10/dist-packages (from nbclassic>=0.4.7->notebook->jupyter==1.0.0->d2l==
1.0.3) (0.2.4)
Requirement already satisfied: fastjsonschema>=2.15 in /usr/local/lib/pytho
n3.10/dist-packages (from nbformat>=5.1->nbconvert->jupyter==1.0.0->d2l==1.
0.3) (2.20.0)
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.1
0/dist-packages (from nbformat>=5.1->nbconvert->jupyter==1.0.0->d2l==1.0.3)
(4.23.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-pa
ckages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->jupyter-console-
>jupyter==1.0.0->d2l==1.0.3) (0.2.13)
Requirement already satisfied: ptyprocess in /usr/local/lib/python3.10/dist
-packages (from terminado>=0.8.3->notebook->jupyter==1.0.0->d2l==1.0.3) (0.
7.0)
Requirement already satisfied: argon2-cffi-bindings in /usr/local/lib/pytho
n3.10/dist-packages (from argon2-cffi->notebook->jupyter==1.0.0->d2l==1.0.
3) (21.2.0)
```

```
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/d
ist-packages (from beautifulsoup4->nbconvert->jupyter==1.0.0->d2l==1.0.3)
(2.6)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/di
st-packages (from bleach->nbconvert->jupyter==1.0.0->d2l==1.0.3) (0.5.1)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python
3.10/dist-packages (from jedi>=0.16->ipython>=5.0.0->ipykernel->jupyter==1.
0.0->d2l==1.0.3) (0.8.4)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/d
ist-packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyter==1.0.
0->d2l==1.0.3) (24.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /us
r/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat>=5.1->
nbconvert->jupyter==1.0.0->d2l==1.0.3) (2023.12.1)
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python
3.10/dist-packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyter
==1.0.0->d2l==1.0.3) (0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/
dist-packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyter==1.
0.0->d2l==1.0.3) (0.20.0)
Requirement already satisfied: jupyter-server<3,>=1.8 in /usr/local/lib/pyt
hon3.10/dist-packages (from notebook-shim>=0.2.3->nbclassic>=0.4.7->noteboo
k->jupyter==1.0.0->d2l==1.0.3) (1.24.0)
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.10/dis
t-packages (from argon2-cffi-bindings->argon2-cffi->notebook->jupyter==1.0.
0->d2l==1.0.3) (1.17.1)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-
packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->notebook->ju
pyter==1.0.0->d2l==1.0.3) (2.22)
Requirement already satisfied: anyio<4,>=3.1.0 in /usr/local/lib/python3.1
0/dist-packages (from jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbclass
ic>=0.4.7->notebook->jupyter==1.0.0->d2l==1.0.3) (3.7.1)
Requirement already satisfied: websocket-client in /usr/local/lib/python3.1
0/dist-packages (from jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbclass
ic>=0.4.7->notebook->jupyter==1.0.0->d2l==1.0.3) (1.8.0)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/di
st-packages (from anyio<4,>=3.1.0->jupyter-server<3,>=1.8->notebook-shim>=
0.2.3->nbclassic>=0.4.7->notebook->jupyter==1.0.0->d2l==1.0.3) (1.3.1)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/
dist-packages (from anyio<4,>=3.1.0->jupyter-server<3,>=1.8->notebook-shim>
=0.2.3->nbclassic>=0.4.7->notebook->jupyter==1.0.0->d2l==1.0.3) (1.2.2)
Downloading d2l-1.0.3-py3-none-any.whl (111 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 111.7/111.7 kB 4.9 MB/s eta 0:0
0:00
Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Downloading matplotlib-3.7.2-cp310-cp310-manylinux_2_17_x86_64.manylinux201
4_x86_64.whl (11.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.6/11.6 MB 28.2 MB/s eta 0:0
0:00
Downloading matplotlib_inline-0.1.6-py3-none-any.whl (9.4 kB)
Downloading numpy-1.23.5-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x8
6_64.whl (17.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.1/17.1 MB 29.7 MB/s eta 0:0
0:00
Downloading pandas-2.0.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x8
6_64.whl (12.3 MB)
```

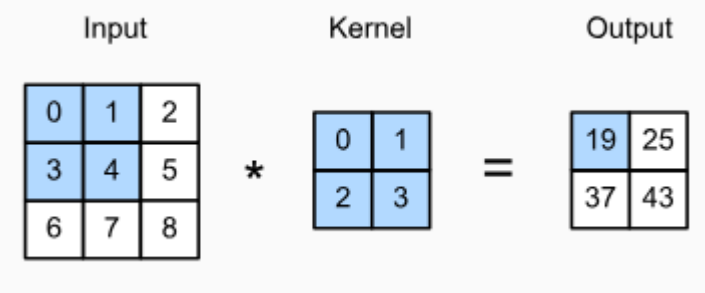
```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.3/12.3 MB 25.0 MB/s eta 0:0
0:00
Downloading requests-2.31.0-py3-none-any.whl (62 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.6/62.6 kB 3.8 MB/s eta 0:00:
00
Downloading scipy-1.10.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x8
6_64.whl (34.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 34.4/34.4 MB 12.0 MB/s eta 0:0
0:00
Downloading pyparsing-3.0.9-py3-none-any.whl (98 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.3/98.3 kB 6.4 MB/s eta 0:00:
00
Downloading qtconsole-5.6.0-py3-none-any.whl (124 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 124.7/124.7 kB 4.0 MB/s eta 0:0
0:00
Downloading QtPy-2.4.1-py3-none-any.whl (93 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 93.5/93.5 kB 4.6 MB/s eta 0:00:
00
Using cached jedi-0.19.1-py2.py3-none-any.whl (1.6 MB)
Installing collected packages: requests, qtpy, pyparsing, numpy, matplotlib
-inline, jedi, scipy, pandas, matplotlib, qtconsole, jupyter, d2l
  Attempting uninstall: requests
    Found existing installation: requests 2.32.3
    Uninstalling requests-2.32.3:
      Successfully uninstalled requests-2.32.3
  Attempting uninstall: pyparsing
    Found existing installation: pyparsing 3.1.4
    Uninstalling pyparsing-3.1.4:
      Successfully uninstalled pyparsing-3.1.4
  Attempting uninstall: numpy
    Found existing installation: numpy 1.26.4
    Uninstalling numpy-1.26.4:
      Successfully uninstalled numpy-1.26.4
  Attempting uninstall: matplotlib-inline
    Found existing installation: matplotlib-inline 0.1.7
    Uninstalling matplotlib-inline-0.1.7:
      Successfully uninstalled matplotlib-inline-0.1.7
  Attempting uninstall: scipy
    Found existing installation: scipy 1.13.1
    Uninstalling scipy-1.13.1:
      Successfully uninstalled scipy-1.13.1
  Attempting uninstall: pandas
    Found existing installation: pandas 2.2.2
    Uninstalling pandas-2.2.2:
      Successfully uninstalled pandas-2.2.2
  Attempting uninstall: matplotlib
    Found existing installation: matplotlib 3.7.1
    Uninstalling matplotlib-3.7.1:
      Successfully uninstalled matplotlib-3.7.1
ERROR: pip's dependency resolver does not currently take into account all t
he packages that are installed. This behaviour is the source of the followi
ng dependency conflicts.
albucore 0.0.16 requires numpy>=1.24, but you have numpy 1.23.5 which is in
compatible.
albumentations 1.4.15 requires numpy>=1.24.4, but you have numpy 1.23.5 whi
ch is incompatible.
```

```
bigframes 1.21.0 requires numpy>=1.24.0, but you have numpy 1.23.5 which is
incompatible.
chex 0.1.87 requires numpy>=1.24.1, but you have numpy 1.23.5 which is inco
mpatible.
google-colab 1.0.0 requires pandas==2.2.2, but you have pandas 2.0.3 which
is incompatible.
google-colab 1.0.0 requires requests==2.32.3, but you have requests 2.31.0
which is incompatible.
jax 0.4.33 requires numpy>=1.24, but you have numpy 1.23.5 which is incompa
tible.
jaxlib 0.4.33 requires numpy>=1.24, but you have numpy 1.23.5 which is inco
mpatible.
mizani 0.11.4 requires pandas>=2.1.0, but you have pandas 2.0.3 which is in
compatible.
plotnine 0.13.6 requires pandas<3.0.0,>=2.1.0, but you have pandas 2.0.3 wh
ich is incompatible.
xarray 2024.9.0 requires numpy>=1.24, but you have numpy 1.23.5 which is in
compatible.
xarray 2024.9.0 requires pandas>=2.1, but you have pandas 2.0.3 which is in
compatible.
Successfully installed d2l-1.0.3 jedi-0.19.1 jupyter-1.0.0 matplotlib-3.7.2
matplotlib-inline-0.1.6 numpy-1.23.5 pandas-2.0.3 pyparsing-3.0.9 qtconsole
-5.6.0 qtpy-2.4.1 requests-2.31.0 scipy-1.10.1
```

Img에 대한 Convolutional

In [3]:
```
import torch
from torch import nn
from d2l import torch as d2l
```

엄밀하게: 교차상관 연산



3 * 3의 tensor에서: 2*2 부분씩 계산

ㄴ 0*0+ 1*1+ 2*3+ 3*4= 19인거

전체 반복#: **(입력크기-kernel크기)/strd +1**= (3-2)/1+1

In [4]:
```
def corr2d(X, K):  #@save
    """Compute 2D cross-correlation."""
    h, w = K.shape
    Y = torch.zeros((X.shape[0] - h + 1, X.shape[1] - w + 1))
    for i in range(Y.shape[0]):
        for j in range(Y.shape[1]):
```

```
            Y[i, j] = (X[i:i + h, j:j + w] * K).sum()
    return Y
```

In [5]:
```
X = torch.tensor([[0.0, 1.0, 2.0], [3.0, 4.0, 5.0], [6.0, 7.0, 8.0]])
K = torch.tensor([[0.0, 1.0], [2.0, 3.0]])
corr2d(X, K)
```

Out[5]:
```
tensor([[19., 25.],
        [37., 43.]])
```

합성곱 계층

두 para: kernel, bias

ㄴ __init__에서 kenrel weight와 bias 초기화

In [6]:
```
class Conv2D(nn.Module):
    def __init__(self, kernel_size):
        super().__init__()
        self.weight = nn.Parameter(torch.rand(kernel_size))
        self.bias = nn.Parameter(torch.zeros(1))

    def forward(self, x):
        return corr2d(x, self.weight) + self.bias
```

img부터 구성해보자: 6 * 8

In [7]:
```
X = torch.ones((6, 8))
X[:, 2:6] = 0 #2~6은 0
X
```

Out[7]:
```
tensor([[1., 1., 0., 0., 0., 0., 1., 1.],
        [1., 1., 0., 0., 0., 0., 1., 1.],
        [1., 1., 0., 0., 0., 0., 1., 1.],
        [1., 1., 0., 0., 0., 0., 1., 1.],
        [1., 1., 0., 0., 0., 0., 1., 1.],
        [1., 1., 0., 0., 0., 0., 1., 1.]])
```

kernel 구성: **w 2, h 1**

In [8]:
```
K = torch.tensor([[1.0, -1.0]])
```

흰->검 가장자리는 1

검->흰은 -1

In [9]:
```
Y = corr2d(X, K)
Y
```

Out[9]:
```
tensor([[ 0.,  1.,  0.,  0.,  0., -1.,  0.],
        [ 0.,  1.,  0.,  0.,  0., -1.,  0.],
        [ 0.,  1.,  0.,  0.,  0., -1.,  0.],
        [ 0.,  1.,  0.,  0.,  0., -1.,  0.],
        [ 0.,  1.,  0.,  0.,  0., -1.,  0.],
        [ 0.,  1.,  0.,  0.,  0., -1.,  0.]])
```

어차피 k입력과 합성곱하면: **수평곱 요소** = 출력 0

In [10]:
```
corr2d(X.t(), K)
```

Out[10]:
```
tensor([[0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]])
```

kernel 학습

입출력 쌍만 보고 학습 가능?

합성곱 layer 구성-> kernel을 tensor로 초기화

단순성: 내장class 사용, bias 무시 -> epoch 10번 하고: 오류 적어짐

In [11]:
```
# Construct a two-dimensional convolutional layer with 1 output channel an
# kernel of shape (1, 2). For the sake of simplicity, we ignore the bias h
conv2d = nn.LazyConv2d(1, kernel_size=(1, 2), bias=False)

# The two-dimensional convolutional layer uses four-dimensional input and
# output in the format of (example, channel, height, width), where the bat
# size (number of examples in the batch) and the number of channels are bo
X = X.reshape((1, 1, 6, 8))
Y = Y.reshape((1, 1, 6, 7))
lr = 3e-2  # Learning rate

for i in range(10):
    Y_hat = conv2d(X)
    l = (Y_hat - Y) ** 2
    conv2d.zero_grad()
    l.sum().backward()
    # Update the kernel
    conv2d.weight.data[:] -= lr * conv2d.weight.grad
    if (i + 1) % 2 == 0:
        print(f'epoch {i + 1}, loss {l.sum():.3f}')
```

```
epoch 2, loss 7.748
epoch 4, loss 1.901
epoch 6, loss 0.565
epoch 8, loss 0.196
epoch 10, loss 0.074
```

실제 우리가 정의한 tensor ([1, -1])과 비슷

In [12]:
```
conv2d.weight.data.reshape((1, 2))
```

Out[12]:
```
tensor([[ 0.9607, -1.0154]])
```

합성곱은 **Feature Map**이라고도 함

ㄴ 후속 계층에 대한 w,h의 학습된 feaure로 간주 가능

Padding and Stride

ex: 입력: 3*3 kernel: 2*2*

Stride로 출력 **제어**해보자!

In [13]:
```
import torch
from torch import nn
```

Padding

합성곱 할때: 모서리 안씀

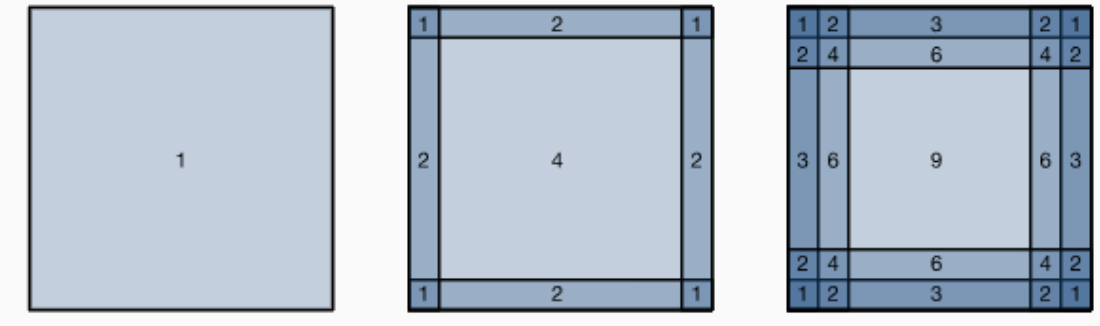


그림 7.3.1 크기의 합성곱에 대한 픽셀 활용$1 \times 1, 2 \times 2$, 그리고3×3각기

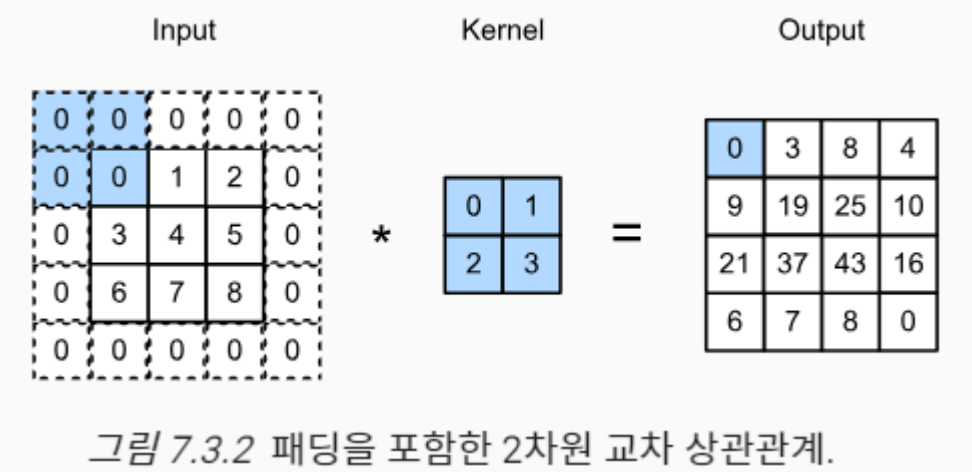


그림 7.3.2 패딩을 포함한 2차원 교차 상관관계.

해결책: 입력 주변에 추가 pixel

ㄴ img의 효과적 크기 늘림

3 * 3-> 5 * 5

In [14]:
```
# We define a helper function to calculate convolutions. It initializes th
# convolutional layer weights and performs corresponding dimensionality
# elevations and reductions on the input and output
def comp_conv2d(conv2d, X):
    # (1, 1) indicates that batch size and the number of channels are both
    X = X.reshape((1, 1) + X.shape)
    Y = conv2d(X)
    # Strip the first two dimensions: examples and channels
    return Y.reshape(Y.shape[2:])

# 1 row and column is padded on either side, so a total of 2 rows or colum
# are added
conv2d = nn.LazyConv2d(1, kernel_size=3, padding=1)
X = torch.rand(size=(8, 8))
comp_conv2d(conv2d, X).shape
```

Out[14]:
```
torch.Size([8, 8])
```

w h 3인 2차원 합성곱 layer-> 1px 패딩

8 * 8이면-> 출력도 8 * 8

In [15]:
```
# We use a convolution kernel with height 5 and width 3. The padding on ei
# side of the height and width are 2 and 1, respectively
conv2d = nn.LazyConv2d(1, kernel_size=(5, 3), padding=(2, 1))
comp_conv2d(conv2d, X).shape
```

Out[15]:
```
torch.Size([8, 8])
```

Stride

때로는 Downsampling<- 중간 건너뜀

In [16]:
```
conv2d = nn.LazyConv2d(1, kernel_size=3, padding=1, stride=2)
comp_conv2d(conv2d, X).shape
```

Out[16]:
```
torch.Size([4, 4])
```

In [17]:
```
conv2d = nn.LazyConv2d(1, kernel_size=(3, 5), padding=(0, 1), stride=(3, 4
comp_conv2d(conv2d, X).shape
```

Out[17]:
```
torch.Size([2, 2])
```

다중 입출력 channel

원래 단일 chan로 했음<- 2차원 tensor로 생각 가능

Mix에 추가하면 3차원!!

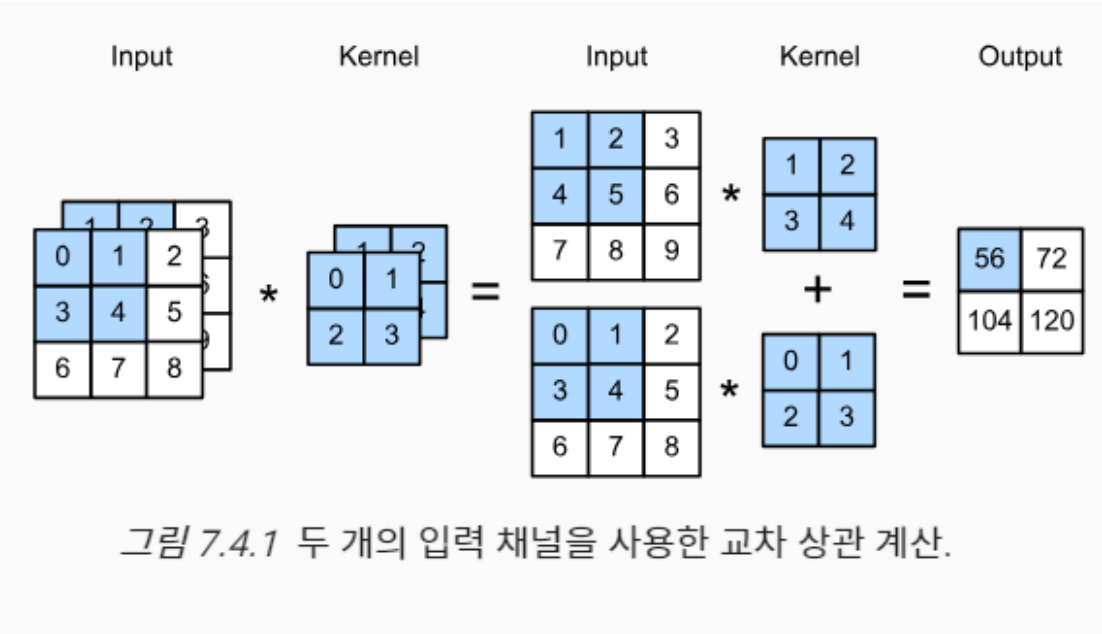


그림 7.4.1 두 개의 입력 채널을 사용한 교차 상관 계산.

각 chan별로 알맞는 kernel있음

In [18]:
```
import torch
from d2l import torch as d2l
```

In [19]:
```
def corr2d_multi_in(X, K):
    # Iterate through the 0th dimension (channel) of K first, then add the
    return sum(d2l.corr2d(x, k) for x, k in zip(X, K))
```

In [20]:
```
X = torch.tensor([[[0.0, 1.0, 2.0], [3.0, 4.0, 5.0], [6.0, 7.0, 8.0]],
                  [[1.0, 2.0, 3.0], [4.0, 5.0, 6.0], [7.0, 8.0, 9.0]]])
K = torch.tensor([[[0.0, 1.0], [2.0, 3.0]], [[1.0, 2.0], [3.0, 4.0]]])

corr2d_multi_in(X, K)
```

Out[20]:
```
tensor([[ 56.,  72.],
        [104., 120.]])
```

아니 출력도 여러 chan?

교차 상관 함수 구현함

output chan: k, k+1, k+2

In [21]:
```
def corr2d_multi_in_out(X, K):
    # Iterate through the 0th dimension of K, and each time, perform
    # cross-correlation operations with input X. All of the results are
    # stacked together
    return torch.stack([corr2d_multi_in(X, k) for k in K], 0)
```

In [22]:
```
K = torch.stack((K, K + 1, K + 2), 0)
K.shape
```

Out[22]:
```
torch.Size([3, 2, 2, 2])
```

In [23]:
```
corr2d_multi_in_out(X, K)
```

Out[23]:
```
tensor([[[ 56.,  72.],
         [104., 120.]],

        [[ 76., 100.],
         [148., 172.]],

        [[ 96., 128.],
         [192., 224.]]])
```

1 * 1 합성곱 layer

최소 창: 더 큰 합성layer는 상실, 대신 chan차원

In [24]:
```
def corr2d_multi_in_out_1x1(X, K):
    c_i, h, w = X.shape
    c_o = K.shape[0]
    X = X.reshape((c_i, h * w))
    K = K.reshape((c_o, c_i))
    # Matrix multiplication in the fully connected layer
    Y = torch.matmul(K, X)
    return Y.reshape((c_o, h, w))
```

In [25]:
```
X = torch.normal(0, 1, (3, 3, 3))
K = torch.normal(0, 1, (2, 3, 1, 1))
Y1 = corr2d_multi_in_out_1x1(X, K)
Y2 = corr2d_multi_in_out(X, K)
assert float(torch.abs(Y1 - Y2).sum()) < 1e-6
```

Pooling: Downsampling, 민감도 완화

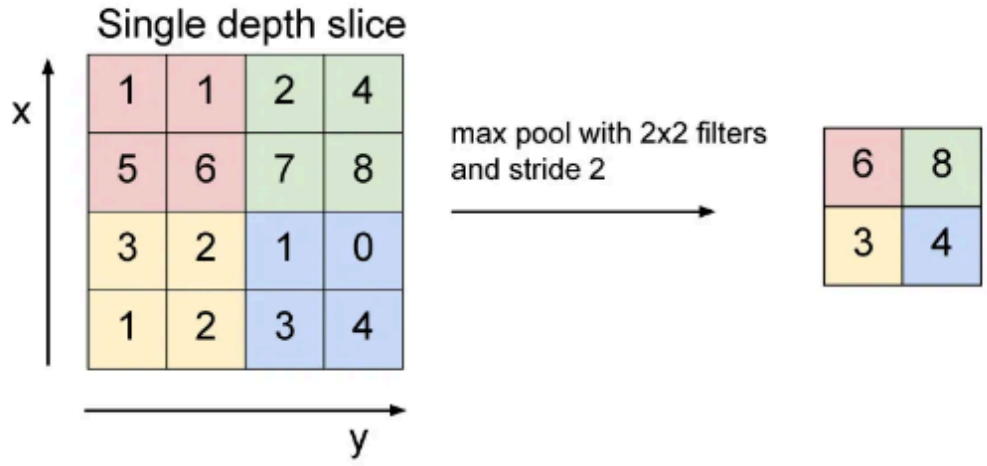
최종 layer: 전체 img 민감해야

모서리 level: 불변하기 위해

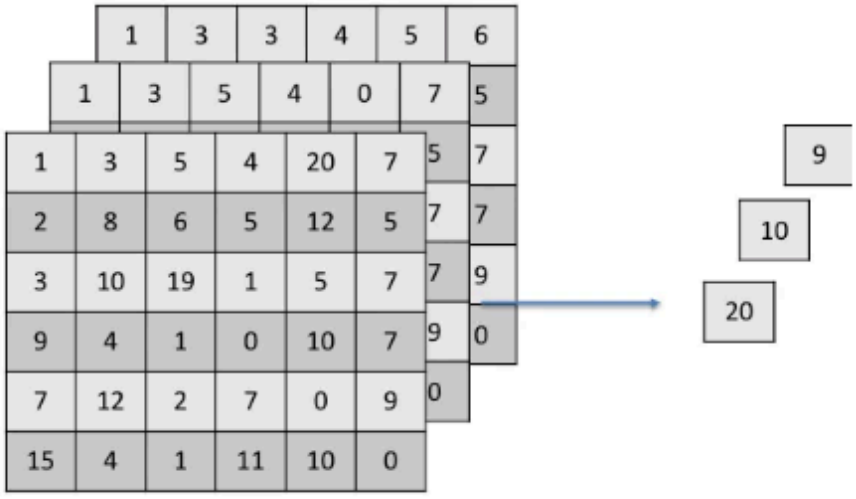
In [26]:
```
import torch
from torch import nn
from d2l import torch as d2l
```

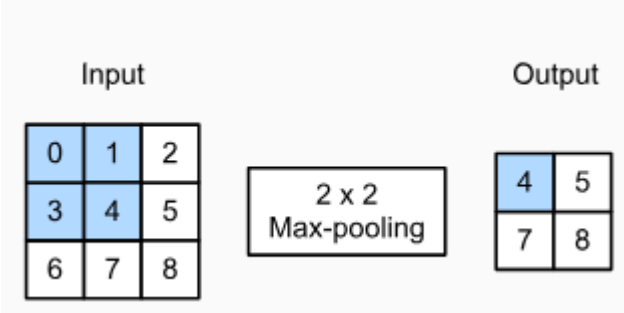
Max, Aver Pooling

- Pool방법: Max, 전역_Aver



Max를 주로 씀





In [27]:
```
def pool2d(X, pool_size, mode='max'):
    p_h, p_w = pool_size
    Y = torch.zeros((X.shape[0] - p_h + 1, X.shape[1] - p_w + 1))
    for i in range(Y.shape[0]):
        for j in range(Y.shape[1]):
            if mode == 'max':
                Y[i, j] = X[i: i + p_h, j: j + p_w].max()
            elif mode == 'avg':
                Y[i, j] = X[i: i + p_h, j: j + p_w].mean()
    return Y
```

In [28]:
```
X = torch.tensor([[0.0, 1.0, 2.0], [3.0, 4.0, 5.0], [6.0, 7.0, 8.0]])
pool2d(X, (2, 2))
```

Out[28]:
```
tensor([[4., 5.],
        [7., 8.]])
```

In [29]:
```
pool2d(X, (2, 2), 'avg')
```

Out[29]:
```
tensor([[2., 3.],
        [5., 6.]])
```

In [30]:
```
X = torch.arange(16, dtype=torch.float32).reshape((1, 1, 4, 4))
X
```

Out[30]:
```
tensor([[[[ 0.,  1.,  2.,  3.],
          [ 4.,  5.,  6.,  7.],
          [ 8.,  9., 10., 11.],
          [12., 13., 14., 15.]]]])
```

3 * 3짜리

In [31]:
```
pool2d = nn.MaxPool2d(3)
# Pooling has no model parameters, hence it needs no initialization
pool2d(X)
```

Out[31]:
```
tensor([[[[10.]]]])
```

그냥 stride 2짜리에요

In [32]:
```
pool2d = nn.MaxPool2d(3, padding=1, stride=2)
pool2d(X)
```

Out[32]:
```
tensor([[[[ 5.,  7.],
          [13., 15.]]]])
```

In [33]:
```
pool2d = nn.MaxPool2d((2, 3), stride=(2, 3), padding=(0, 1))
pool2d(X)
```

Out[33]:
```
tensor([[[[ 5.,  7.],
          [13., 15.]]]])
```

다중채널 .cat((X,X+1),1)

Pooling해도 chan은 2개

In [34]:
```
X = torch.cat((X, X + 1), 1)
X
```

Out[34]:
```
tensor([[[[ 0.,  1.,  2.,  3.],
          [ 4.,  5.,  6.,  7.],
          [ 8.,  9., 10., 11.],
          [12., 13., 14., 15.]],

         [[ 1.,  2.,  3.,  4.],
          [ 5.,  6.,  7.,  8.],
          [ 9., 10., 11., 12.],
          [13., 14., 15., 16.]]]])
```

In [35]:
```
pool2d = nn.MaxPool2d(3, padding=1, stride=2)
pool2d(X)
```

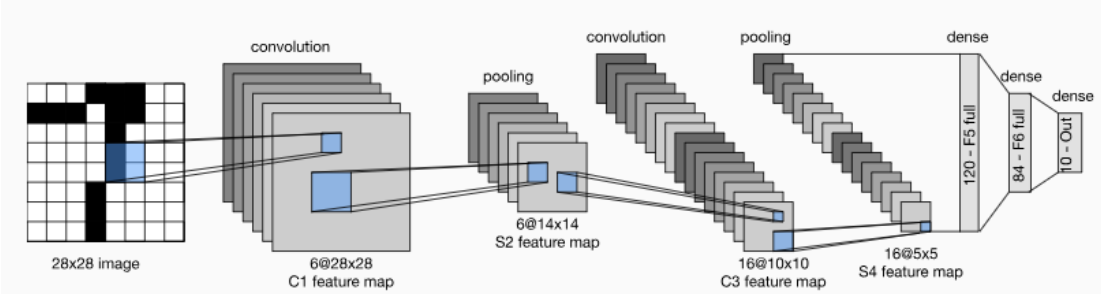
Out[35]:
```
tensor([[[[ 5.,  7.],
          [13., 15.]],

         [[ 6.,  8.],
          [14., 16.]]]])
```

Lenet

전에 softmax와 MLP로 Linear-Model 구현-> 의류사진에 적용

합성곱: 적은 변수

In [36]:
```
import torch
from torch import nn
from d2l import torch as d2l
```



각 block 기본 단위: 합성곱, sigmmoid, aver_pooling

가우시안-> softmax

2 * 2 pooling이 차원 1/2배

합성곱 layer별 출력chan#:

1. 6
2. 16

In [37]:
```
def init_cnn(module):  #@save
    """Initialize weights for CNNs."""
```

```
    if type(module) == nn.Linear or type(module) == nn.Conv2d:
        nn.init.xavier_uniform_(module.weight)
```

In [38]:
```
class LeNet(d2l.Classifier):  #@save
    """The LeNet-5 model."""
    def __init__(self, lr=0.1, num_classes=10):
        super().__init__()
        self.save_hyperparameters()
        self.net = nn.Sequential(
            nn.LazyConv2d(6, kernel_size=5, padding=2), nn.Sigmoid(),
            nn.AvgPool2d(kernel_size=2, stride=2),
            nn.LazyConv2d(16, kernel_size=5), nn.Sigmoid(),
            nn.AvgPool2d(kernel_size=2, stride=2),
            nn.Flatten(),
            nn.LazyLinear(120), nn.Sigmoid(),
            nn.LazyLinear(84), nn.Sigmoid(),
            nn.LazyLinear(num_classes))
```

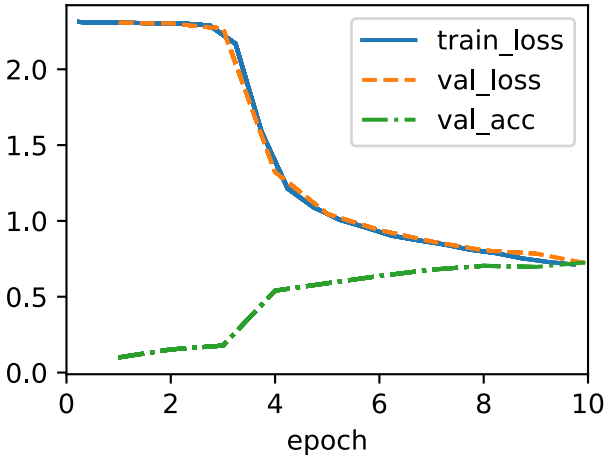
In [39]:
```
@d2l.add_to_class(d2l.Classifier)  #@save
def layer_summary(self, X_shape):
    X = torch.randn(*X_shape)
    for layer in self.net:
        X = layer(X)
        print(layer.__class__.__name__, 'output shape:\t', X.shape)

model = LeNet()
model.layer_summary((1, 1, 28, 28))
```

```
Conv2d output shape:     torch.Size([1, 6, 28, 28])
Sigmoid output shape:    torch.Size([1, 6, 28, 28])
AvgPool2d output shape:  torch.Size([1, 6, 14, 14])
Conv2d output shape:     torch.Size([1, 16, 10, 10])
Sigmoid output shape:    torch.Size([1, 16, 10, 10])
AvgPool2d output shape:  torch.Size([1, 16, 5, 5])
Flatten output shape:    torch.Size([1, 400])
Linear output shape:     torch.Size([1, 120])
Sigmoid output shape:    torch.Size([1, 120])
Linear output shape:     torch.Size([1, 84])
Sigmoid output shape:    torch.Size([1, 84])
Linear output shape:     torch.Size([1, 10])
```

Training

para는 적지만, 더 많은 곱셈 참여-> MLP보다 비쌈

In [40]:
```
trainer = d2l.Trainer(max_epochs=10, num_gpus=1)
data = d2l.FashionMNIST(batch_size=128)
model = LeNet(lr=0.1)
model.apply_init([next(iter(data.get_dataloader(True)))[0]], init_cnn)
trainer.fit(model, data)
```



Networks Using Blocks (VGG)

:label: `sec_vgg`

While AlexNet offered empirical evidence that deep CNNs can achieve good results, it did not provide a general template to guide subsequent researchers in designing new networks. In the following sections, we will introduce several heuristic concepts commonly used to design deep networks.

Progress in this field mirrors that of VLSI (very large scale integration) in chip design where engineers moved from placing transistors to logical elements to logic blocks :cite: `Mead.1980` . Similarly, the design of neural network architectures has grown progressively more abstract, with researchers moving from thinking in terms of individual neurons to whole layers, and now to blocks, repeating patterns of layers. A decade later, this has now progressed to researchers using entire trained models to repurpose them for different, albeit related, tasks. Such large pretrained models are typically called *foundation models* :cite: `bommasani2021opportunities` .

Back to network design. The idea of using blocks first emerged from the Visual Geometry Group (VGG) at Oxford University, in their eponymously-named *VGG* network :cite: `Simonyan.Zisserman.2014` . It is easy to implement these repeated structures in code with any modern deep learning framework by using loops and subroutines.

In [41]:
```
import torch
from torch import nn
from d2l import torch as d2l
```

VGG: Block사용 net

block: layer의 반복패턴

=칩설계에서 VLSI 발전과 유사

Transister배치->Logic배치->논리block

NN: 점점 더 추상

개별Neuron-> 전체layer-> lyaer반복패턴(Block)

CNN 기본 구성 요소:

1. 합성곱(FOR 해상도)
2. ReLU(비선형성)
3. Max Pooling ㄴ해상도 너무 빨리 줄음

->해결책: Block으로 downsampling에 합성곱 사용

깊고 좁은 net이 낫다-> 100 layer 이상

In [42]:
```
def vgg_block(num_convs, out_channels):
    layers = []
    for _ in range(num_convs):
        layers.append(nn.LazyConv2d(out_channels, kernel_size=3, padding=1
        layers.append(nn.ReLU())
    layers.append(nn.MaxPool2d(kernel_size=2,stride=2))
    return nn.Sequential(*layers)
```

VGG

1. 합성곱 + Pooling
2. 완전 연결(=AlexNet)

In [43]:
```
class VGG(d2l.Classifier):
    def __init__(self, arch, lr=0.1, num_classes=10):
        super().__init__()
        self.save_hyperparameters()
        conv_blks = []
        for (num_convs, out_channels) in arch:
            conv_blks.append(vgg_block(num_convs, out_channels))
        self.net = nn.Sequential(
            *conv_blks, nn.Flatten(),
            nn.LazyLinear(4096), nn.ReLU(), nn.Dropout(0.5),
            nn.LazyLinear(4096), nn.ReLU(), nn.Dropout(0.5),
            nn.LazyLinear(num_classes))
        self.net.apply(d2l.init_cnn)
```

원래 5개의 합성곱 block

1. 64개
2. 이후 2배씩-> 512개

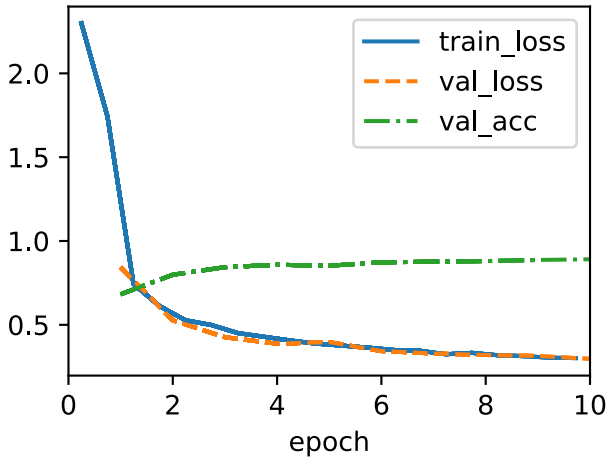
In [44]:
```
VGG(arch=((1, 64), (1, 128), (2, 256), (2, 512), (2, 512))).layer_summary(
    (1, 1, 224, 224))
```

```
Sequential output shape:         torch.Size([1, 64, 112, 112])
Sequential output shape:         torch.Size([1, 128, 56, 56])
Sequential output shape:         torch.Size([1, 256, 28, 28])
Sequential output shape:         torch.Size([1, 512, 14, 14])
Sequential output shape:         torch.Size([1, 512, 7, 7])
Flatten output shape:    torch.Size([1, 25088])
Linear output shape:     torch.Size([1, 4096])
ReLU output shape:       torch.Size([1, 4096])
Dropout output shape:    torch.Size([1, 4096])
Linear output shape:     torch.Size([1, 4096])
ReLU output shape:       torch.Size([1, 4096])
Dropout output shape:    torch.Size([1, 4096])
Linear output shape:     torch.Size([1, 10])
```

Training

chan수 적게 구성

In [45]:
```
model = VGG(arch=((1, 16), (1, 32), (2, 64), (2, 128), (2, 128)), lr=0.01)
trainer = d2l.Trainer(max_epochs=10, num_gpus=1)
data = d2l.FashionMNIST(batch_size=128, resize=(224, 224))
model.apply_init([next(iter(data.get_dataloader(True)))[0]], d2l.init_cnn)
trainer.fit(model, data)
```



ResNet: 잔여net

net이 복잡해지면 어떻게 돼?

Layer 추가하면 net표현력 풍부해짐

In [46]:
```
import torch
from torch import nn
from torch.nn import functional as F
from d2l import torch as d2l
```

함수 class

정규화: 복잡성 제어 가능

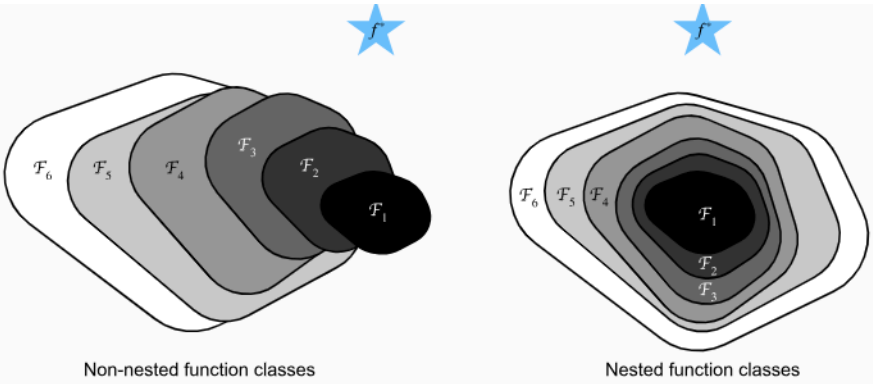


그림 8.6.1 중첩되지 않은 함수 클래스의 경우 더 큰(면적으로 표시됨) 함수 클래스는 우리가 "진실" 함수에 더 가까워질 것이라는 것을 보장하지 않습니다.(f^*). 이런 일은 중첩된 함수 클래스에서는 일어나지 않습니다.

라서 더 큰 함수 클래스에 더 작은 함수 클래스가 포함되어 있는 경우에만 이를 증가시키면 네트워크의 표현력 | 엄격하게 증가한다는 것이 보장됩니다. 딥 신경망의 경우 새로 추가된 레이어를 항등 함수로 학습할 수 있다 $f(\mathbf{x}) = \mathbf{x}$, 새로운 모델은 원래 모델만큼 효과적일 것입니다. 새로운 모델이 훈련 데이터 세트에 맞는 더 나 솔루션을 얻을 수 있으므로 추가된 레이어로 인해 훈련 오류를 줄이는 것이 더 쉬워질 수 있습니다.

중첩x 함수class만: **값 가까워질 것** 보장

In [47]:
```
class Residual(nn.Module):  #@save
    """The Residual block of ResNet models."""
    def __init__(self, num_channels, use_1x1conv=False, strides=1):
        super().__init__()
        self.conv1 = nn.LazyConv2d(num_channels, kernel_size=3, padding=1,
                                   stride=strides)
        self.conv2 = nn.LazyConv2d(num_channels, kernel_size=3, padding=1)
        if use_1x1conv:
            self.conv3 = nn.LazyConv2d(num_channels, kernel_size=1,
                                       stride=strides)
        else:
            self.conv3 = None
        self.bn1 = nn.LazyBatchNorm2d()
        self.bn2 = nn.LazyBatchNorm2d()

    def forward(self, X):
        Y = F.relu(self.bn1(self.conv1(X)))
        Y = self.bn2(self.conv2(Y))
        if self.conv3:
            X = self.conv3(X)
        Y += X
        return F.relu(Y)
```

입출력 =

In [48]:
```
blk = Residual(3)
X = torch.randn(4, 3, 6, 6)
blk(X).shape
```

Out[48]:
```
torch.Size([4, 3, 6, 6])
```

출력chan 늘리고, 출력h w 반으로

In [49]:
```
blk = Residual(6, use_1x1conv=True, strides=2)
blk(X).shape
```

Out[49]:
```
torch.Size([4, 6, 3, 3])
```

ResNet

=GoogleNet

7 * 764 출력 합성곱: stride 2

Pooling: 3 * 3, 2stride

In [50]:
```
class ResNet(d2l.Classifier):
    def b1(self):
        return nn.Sequential(
            nn.LazyConv2d(64, kernel_size=7, stride=2, padding=3),
            nn.LazyBatchNorm2d(), nn.ReLU(),
            nn.MaxPool2d(kernel_size=3, stride=2, padding=1))
```

ResNet: 잔여 block 4개모듈

첫 모듈: max_Pool 사용함-> h w 줄일필요x

후속 모듈: chan# 2배, h w 절반

In [51]:
```
@d2l.add_to_class(ResNet)
def block(self, num_residuals, num_channels, first_block=False):
    blk = []
    for i in range(num_residuals):
        if i == 0 and not first_block:
            blk.append(Residual(num_channels, use_1x1conv=True, strides=2))
        else:
            blk.append(Residual(num_channels))
    return nn.Sequential(*blk)
```

ResNet에 추가

In [52]:
```
@d2l.add_to_class(ResNet)
def __init__(self, arch, lr=0.1, num_classes=10):
    super(ResNet, self).__init__()
    self.save_hyperparameters()
    self.net = nn.Sequential(self.b1())
    for i, b in enumerate(arch):
        self.net.add_module(f'b{i+2}', self.block(*b, first_block=(i==0)))
    self.net.add_module('last', nn.Sequential(
        nn.AdaptiveAvgPool2d((1, 1)), nn.Flatten(),
        nn.LazyLinear(num_classes)))
    self.net.apply(d2l.init_cnn)
```

4개의 합성곱layer

In [53]:
```
class ResNet18(ResNet):
    def __init__(self, lr=0.1, num_classes=10):
        super().__init__(((2, 64), (2, 128), (2, 256), (2, 512)),
                         lr, num_classes)
```

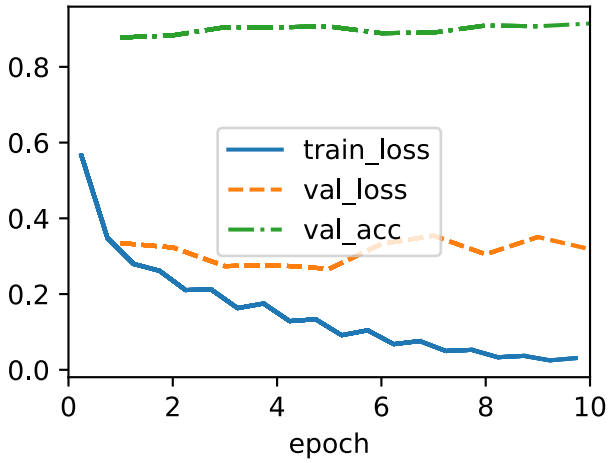
In [54]:
```
ResNet18().layer_summary((1, 1, 96, 96))
```

```
Sequential output shape:         torch.Size([1, 64, 24, 24])
Sequential output shape:         torch.Size([1, 64, 24, 24])
Sequential output shape:         torch.Size([1, 128, 12, 12])
Sequential output shape:         torch.Size([1, 256, 6, 6])
Sequential output shape:         torch.Size([1, 512, 3, 3])
Sequential output shape:         torch.Size([1, 10])
```

Training

train loss 낮음

In [55]:
```
model = ResNet18(lr=0.01)
trainer = d2l.Trainer(max_epochs=10, num_gpus=1)
data = d2l.FashionMNIST(batch_size=128, resize=(96, 96))
model.apply_init([next(iter(data.get_dataloader(True)))[0]], d2l.init_cnn)
trainer.fit(model, data)
```



In [57]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [58]:
```
!pip install --upgrade lxml jinja2 nbconvert
```

```
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packa
ges (4.9.4)
Collecting lxml
  Downloading lxml-5.3.0-cp310-cp310-manylinux_2_28_x86_64.whl.metadata (3.
8 kB)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-pac
kages (3.1.4)
Requirement already satisfied: nbconvert in /usr/local/lib/python3.10/dist-
packages (6.5.4)
Collecting nbconvert
  Downloading nbconvert-7.16.4-py3-none-any.whl.metadata (8.5 kB)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.1
0/dist-packages (from jinja2) (2.1.5)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/
dist-packages (from nbconvert) (4.12.3)
Requirement already satisfied: bleach!=5.0.0 in /usr/local/lib/python3.10/d
ist-packages (from nbconvert) (6.1.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.10/dist
-packages (from nbconvert) (0.7.1)
Requirement already satisfied: jupyter-core>=4.7 in /usr/local/lib/python3.
10/dist-packages (from nbconvert) (5.7.2)
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python
3.10/dist-packages (from nbconvert) (0.3.0)
Collecting mistune<4,>=2.0.3 (from nbconvert)
  Downloading mistune-3.0.2-py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.1
0/dist-packages (from nbconvert) (0.10.0)
Requirement already satisfied: nbformat>=5.7 in /usr/local/lib/python3.10/d
ist-packages (from nbconvert) (5.10.4)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from nbconvert) (24.1)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/pytho
n3.10/dist-packages (from nbconvert) (1.5.1)
Requirement already satisfied: pygments>=2.4.1 in /usr/local/lib/python3.1
0/dist-packages (from nbconvert) (2.18.0)
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.10/dist-p
ackages (from nbconvert) (1.3.0)
Requirement already satisfied: traitlets>=5.1 in /usr/local/lib/python3.10/
dist-packages (from nbconvert) (5.7.1)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.10/dist
-packages (from bleach!=5.0.0->nbconvert) (1.16.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/di
st-packages (from bleach!=5.0.0->nbconvert) (0.5.1)
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.
10/dist-packages (from jupyter-core>=4.7->nbconvert) (4.3.6)
Requirement already satisfied: jupyter-client>=6.1.12 in /usr/local/lib/pyt
hon3.10/dist-packages (from nbclient>=0.5.0->nbconvert) (6.1.12)
Requirement already satisfied: fastjsonschema>=2.15 in /usr/local/lib/pytho
n3.10/dist-packages (from nbformat>=5.7->nbconvert) (2.20.0)
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.1
0/dist-packages (from nbformat>=5.7->nbconvert) (4.23.0)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/d
ist-packages (from beautifulsoup4->nbconvert) (2.6)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/d
ist-packages (from jsonschema>=2.6->nbformat>=5.7->nbconvert) (24.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /us
```

```
r/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat>=5.7->
nbconvert) (2023.12.1)
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python
3.10/dist-packages (from jsonschema>=2.6->nbformat>=5.7->nbconvert) (0.35.
1)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/
dist-packages (from jsonschema>=2.6->nbformat>=5.7->nbconvert) (0.20.0)
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.10/dist-
packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert) (24.0.1)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/pytho
n3.10/dist-packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconver
t) (2.8.2)
Requirement already satisfied: tornado>=4.1 in /usr/local/lib/python3.10/di
st-packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert) (6.3.
3)
Downloading lxml-5.3.0-cp310-cp310-manylinux_2_28_x86_64.whl (5.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.0/5.0 MB 39.9 MB/s eta 0:00:0
0
Downloading nbconvert-7.16.4-py3-none-any.whl (257 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 257.4/257.4 kB 19.2 MB/s eta 0:
00:00
Downloading mistune-3.0.2-py3-none-any.whl (47 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 48.0/48.0 kB 4.3 MB/s eta 0:00:
00
Installing collected packages: mistune, lxml, nbconvert
  Attempting uninstall: mistune
    Found existing installation: mistune 0.8.4
    Uninstalling mistune-0.8.4:
      Successfully uninstalled mistune-0.8.4
  Attempting uninstall: lxml
    Found existing installation: lxml 4.9.4
    Uninstalling lxml-4.9.4:
      Successfully uninstalled lxml-4.9.4
  Attempting uninstall: nbconvert
    Found existing installation: nbconvert 6.5.4
    Uninstalling nbconvert-6.5.4:
      Successfully uninstalled nbconvert-6.5.4
Successfully installed lxml-5.3.0 mistune-3.0.2 nbconvert-7.16.4
```

In [59]:
```
!jupyter nbconvert --to html --template lab /content/drive/MyDrive/Colab\
```

```
[NbConvertApp] Converting notebook /content/drive/MyDrive/Colab Notebooks/k
_univ/COSE242_HW2.ipynb to html
/usr/local/share/jupyter/nbconvert/templates/base/display_priority.j2:32: U
serWarning: Your element with mimetype(s) dict_keys(['application/vnd.colab
-display-data+json']) is not able to be represented.
  {%- elif type == 'text/vnd.mermaid' -%}
[NbConvertApp] WARNING | Alternative text is missing on 1 image(s).
[NbConvertApp] Writing 847896 bytes to /content/drive/MyDrive/Colab Noteboo
ks/k_univ/COSE242_HW2.html
```