7.1 From Fully Connected Layers to Convolutions

In [ ]:
```
%pip install d2l==1.0.3
```

```
Collecting d2l==1.0.3
  Downloading d2l-1.0.3-py3-none-any.whl.metadata (556 bytes)
Collecting jupyter==1.0.0 (from d2l==1.0.3)
  Downloading jupyter-1.0.0-py2.py3-none-any.whl.metadata (995 bytes)
Collecting numpy==1.23.5 (from d2l==1.0.3)
  Downloading numpy-1.23.5-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl.metadata (2.3 kB)
Collecting matplotlib==3.7.2 (from d2l==1.0.3)
  Downloading matplotlib-3.7.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x8
6_64.whl.metadata (5.6 kB)
Collecting matplotlib-inline==0.1.6 (from d2l==1.0.3)
  Downloading 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_6
4.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_6
4.whl.metadata (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.9/58.9 kB 1.7 MB/s eta 0:00:00
Requirement already satisfied: notebook in /usr/local/lib/python3.10/dist-package
s (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/python3.10/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-packag
es (from jupyter==1.0.0->d2l==1.0.3) (6.5.4)
Requirement already satisfied: ipykernel in /usr/local/lib/python3.10/dist-packag
es (from jupyter==1.0.0->d2l==1.0.3) (5.5.6)
Requirement already satisfied: ipywidgets in /usr/local/lib/python3.10/dist-packa
ges (from jupyter==1.0.0->d2l==1.0.3) (7.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/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/dist-pac
kages (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/dis
t-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/dis
t-packages (from matplotlib==3.7.2->d2l==1.0.3) (1.4.7)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/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/dist-pa
ckages (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/python3.10/
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-packag
es (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/dist-pac
kages (from pandas==2.0.3->d2l==1.0.3) (2024.2)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-p
ackages (from pandas==2.0.3->d2l==1.0.3) (2024.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python
3.10/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/dist-pac
kages (from requests==2.31.0->d2l==1.0.3) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/di
```

```
st-packages (from requests==2.31.0->d2l==1.0.3) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/di
st-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-package
s (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/python3.10/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-p
ackages (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-p
ackages (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/dist-pac
kages (from ipykernel->jupyter==1.0.0->d2l==1.0.3) (6.3.3)
Requirement already satisfied: widgetsnbextension~=3.6.0 in /usr/local/lib/python
3.10/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/python
3.10/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 /u
sr/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-package
s (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-packages (f
rom nbconvert->jupyter==1.0.0->d2l==1.0.3) (4.9.4)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-p
ackages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (4.12.3)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages
(from nbconvert->jupyter==1.0.0->d2l==1.0.3) (6.1.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.10/dist-packa
ges (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (0.7.1)
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.10/di
st-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/dist-pack
ages (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/dis
t-packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (5.7.2)
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.10/d
ist-packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (0.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/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/dis
t-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/python3.10/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/dist-pa
ckages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (5.10.4)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.10/
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-package
s (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/dist-pa
ckages (from notebook->jupyter==1.0.0->d2l==1.0.3) (24.0.1)
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.10/dist-pack
ages (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/dis
t-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/dis
t-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/python3.10/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/dis
t-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/python3.10/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/python3.10/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-packag
es (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/dist-pack
ages (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-package
s (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/dist-pack
ages (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/dis
t-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/python3.10/
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/python3.10/
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/python3.10/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-packages
(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-packa
ges (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/python3.10/
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/dist-pa
ckages (from beautifulsoup4->nbconvert->jupyter==1.0.0->d2l==1.0.3) (2.6)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-pac
kages (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/python3.10/d
ist-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/dist-pa
ckages (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 /usr/loca
l/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert->j
upyter==1.0.0->d2l==1.0.3) (2023.12.1)
Requirement already satisfied: referencing>=0.28.4 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) (0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-p
ackages (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/python3.1
0/dist-packages (from notebook-shim>=0.2.3->nbclassic>=0.4.7->notebook->jupyter==
1.0.0->d2l==1.0.3) (1.24.0)
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.10/dist-pack
```

```
ages (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-packag
es (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->notebook->jupyter==1.0.0
->d2l==1.0.3) (2.22)
Requirement already satisfied: anyio<4,>=3.1.0 in /usr/local/lib/python3.10/dist-
packages (from jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbclassic>=0.4.7->no
tebook->jupyter==1.0.0->d2l==1.0.3) (3.7.1)
Requirement already satisfied: websocket-client in /usr/local/lib/python3.10/dist
-packages (from jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbclassic>=0.4.7->n
otebook->jupyter==1.0.0->d2l==1.0.3) (1.8.0)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-pac
kages (from anyio<4,>=3.1.0->jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbclas
sic>=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-p
ackages (from anyio<4,>=3.1.0->jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbcl
assic>=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 4.6 MB/s eta 0:00:00
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.manylinux2014_x86_
64.whl (11.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.6/11.6 MB 77.7 MB/s eta 0:00:00
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_x86_64.w
hl (17.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.1/17.1 MB 72.3 MB/s eta 0:00:00
Downloading pandas-2.0.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.w
hl (12.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.3/12.3 MB 75.4 MB/s eta 0:00:00
Downloading requests-2.31.0-py3-none-any.whl (62 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.6/62.6 kB 4.6 MB/s eta 0:00:00
Downloading scipy-1.10.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.w
hl (34.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 34.4/34.4 MB 16.0 MB/s eta 0:00:00
Downloading pyparsing-3.0.9-py3-none-any.whl (98 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.3/98.3 kB 6.1 MB/s eta 0:00:00
Downloading qtconsole-5.6.0-py3-none-any.whl (124 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 124.7/124.7 kB 7.3 MB/s eta 0:00:00
Downloading QtPy-2.4.1-py3-none-any.whl (93 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 93.5/93.5 kB 7.9 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-inlin
e, 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 the pac
kages that are installed. This behaviour is the source of the following dependenc
y conflicts.
albucore 0.0.16 requires numpy>=1.24, but you have numpy 1.23.5 which is incompat
ible.
albumentations 1.4.15 requires numpy>=1.24.4, but you have numpy 1.23.5 which is
incompatible.
bigframes 1.21.0 requires numpy>=1.24.0, but you have numpy 1.23.5 which is incom
patible.
chex 0.1.87 requires numpy>=1.24.1, but you have numpy 1.23.5 which is incompatib
le.
google-colab 1.0.0 requires pandas==2.2.2, but you have pandas 2.0.3 which is inc
ompatible.
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 incompatible.
jaxlib 0.4.33 requires numpy>=1.24, but you have numpy 1.23.5 which is incompatib
le.
mizani 0.11.4 requires pandas>=2.1.0, but you have pandas 2.0.3 which is incompat
ible.
plotnine 0.13.6 requires pandas<3.0.0,>=2.1.0, but you have pandas 2.0.3 which is
incompatible.
xarray 2024.9.0 requires numpy>=1.24, but you have numpy 1.23.5 which is incompat
ible.
xarray 2024.9.0 requires pandas>=2.1, but you have pandas 2.0.3 which is incompat
ible.
Successfully installed d2l-1.0.3 jedi-0.19.1 jupyter-1.0.0 matplotlib-3.7.2 matpl
otlib-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
```

1)One of the reason that we use CNN is translation invariance and locality of convolutional layer.

7.1.6 Exercices

1. If convolution kernel is delta = 0 , then weight of convolutional kernel is all 0. So the MLP only have the bias value independently for set of channels.

2. A. In case of ASR. 2. Since audio is one dimesional input, the convolution operation will be one dimensional matrix operations. 3. If we get audio input as spectrogram of the audio then we can treat the input file as image. It makes same tools to be able to used a computer vision.
3. As video, if the sequential order is important, then I think the translation invariance can be interupting factor for training.

7.2 Convolutions for Images

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

In [ ]:
```
def corr2d(X, K):
    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 [ ]:
```
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[ ]:
```
tensor([[19., 25.],
        [37., 43.]])
```

In [ ]:
```
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
```

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

Out[ ]:
```
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.]])
```

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

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

Out[ ]:
```
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.]])
```

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

Out[ ]:
```
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.]])
```

In [ ]:
```
conv2d = nn.LazyConv2d(1, kernel_size=(1,2), bias=False)

X = X.reshape((1, 1, 6, 8))
Y = Y.reshape((1, 1, 6, 7))
lr = 3e-2

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

```
epoch 2, loss  5.931
epoch 4, loss  1.074
epoch 6, loss  0.213
epoch 8, loss  0.049
epoch 10, loss  0.014
```

```
c:\Users\tprma\miniconda3\envs\d2l\lib\site-packages\torch\nn\modules\lazy.py:180: UserWarning: Lazy modules are a new feature under heavy development so changes to the API or functionality can happen at any moment.
  warnings.warn('Lazy modules are a new feature under heavy development '
```

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

Out[ ]:
```
tensor([[ 0.9976, -0.9778]])
```

1. Optimized kernel tensor is remarkably close to the kernel tensor K we defined earlier -> Why?

7.2.8 Exercises

3. Lazy modules are a new feature under heavy development -> It means that because lazy module are under developing some functions can be restricted.

7.3 Padding an Stride

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

In [ ]:
```
def comp_conv2d(conv2d, X):
    X = X.reshape((1, 1) + X.shape)
    Y = conv2d(X)
```

```
    return Y.reshape(Y.shape[2:])

conv2d = nn.LazyConv2d(1, kernel_size=3, padding=1)
X = torch.rand(size=(8,8))
comp_conv2d(conv2d, X).shape
```

```
c:\Users\tprma\miniconda3\envs\d2l\lib\site-packages\torch\nn\modules\lazy.py:18
0: UserWarning: Lazy modules are a new feature under heavy development so changes
to the API or functionality can happen at any moment.
  warnings.warn('Lazy modules are a new feature under heavy development '
```

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

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

```
c:\Users\tprma\miniconda3\envs\d2l\lib\site-packages\torch\nn\modules\lazy.py:18
0: UserWarning: Lazy modules are a new feature under heavy development so changes
to the API or functionality can happen at any moment.
  warnings.warn('Lazy modules are a new feature under heavy development '
```

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

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

```
c:\Users\tprma\miniconda3\envs\d2l\lib\site-packages\torch\nn\modules\lazy.py:18
0: UserWarning: Lazy modules are a new feature under heavy development so changes
to the API or functionality can happen at any moment.
  warnings.warn('Lazy modules are a new feature under heavy development '
```

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

7.3.4 Exercises

4. If the stride is bigger than 1, then the size of map decrease and also computational volume does.

5. I think it can extract abstracted feature well.

7.4 Multiple Input and Multiple Output Channels

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

In [ ]:
```
def corr2d_multi_in(X, K):
    return sum(d2l.corr2d(X, K) for X, K in zip(X,K))
```

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

In [ ]:
```
def corr2d_multi_in_out(X, k):
    return torch.stack([corr2d_multi_in(X, k) for k in k], 0)
```

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

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

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

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

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

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

In [ ]:
```
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))
    Y = torch.matmul(K, X)
    return Y.reshape((c_o, h, w))
```

In [ ]:
```
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())
```

7.4.5 Exercises

1. A. If there are no nonlinearity in between, a single convolution k2 * k1 can induce the result of the operatio.

2. The result of multi-opertaion of small filter is equivalent with single operation of big filter.

6. I think that we can design the kernel parallelly is preferable for the engineers.

7.5 Pooling

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

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

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

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

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

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

In [ ]:
```
pool2d = nn.MaxPool2d(3)
pool2d(X)
```

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

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

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

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

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

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

Out[ ]:
```
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.]],

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

         [[ 2.,  3.,  4.,  5.],
          [ 6.,  7.,  8.,  9.],
          [10., 11., 12., 13.],
          [14., 15., 16., 17.]]]])
```

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

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

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

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

         [[ 7.,  9.],
          [15., 17.]]]])
```

However, unlike the cross-correlation computation of the inputs and kernels in the convolutional layer, the pooling layer contains no parameters (there is no kernel). Instead, pooling operators are deterministic, typically calculating either the maximum or the average value of the elements in the pooling window.

7.5.5 Exercises

5. By iterating pooling, max-pooling will converge to the max value result and average pooling will converge to the average value output.

6. Pooling operation is non-linear itself so softmax operation is not needed that much.

7.6 Convolutional Neural Networks(LeNet)

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

In [ ]:
```
def init_cnn(module):
    if type(module) == nn.Linear or type(module) == nn.Conv2d:
        nn.init.xavier_uniform_(module.weight)

class LeNet(d2l.Classifier):
    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)
        )
```

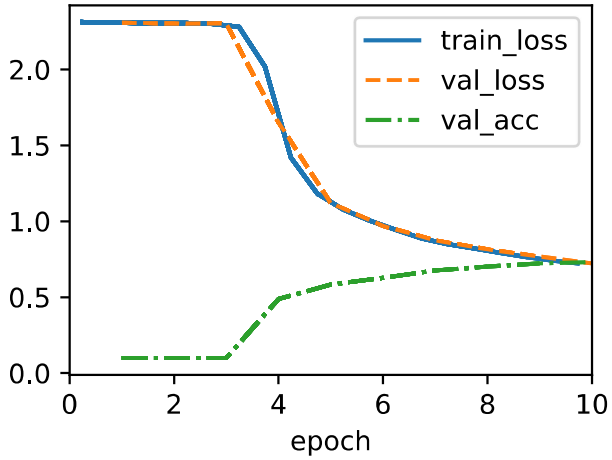
In [ ]:
```
@d2l.add_to_class(d2l.Classifier)
def layer_summary(self, X_shape):
    X = torch.randn(*X_shape)
    for layer in self.net:
        X = layer(X)
        print(layer.__class__.__name__, 'output shape:\t', X.shape)

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])
```

```
c:\Users\tprma\miniconda3\envs\d2l\lib\site-packages\torch\nn\modules\lazy.py:180: UserWarning: Lazy modules are a new feature under heavy development so changes to the API or functionality can happen at any moment.
  warnings.warn('Lazy modules are a new feature under heavy development '
```

In [ ]:
```
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)
```



8.2 Networks Using Block(VGG)

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

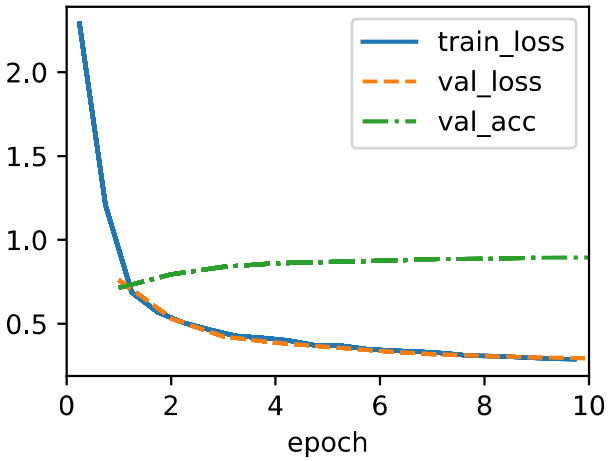
In [ ]:
```
def vgg_block(num_convs, out_channels):
    layers = []
    for _ in range(num_convs):
        layers.append(nn.LazyConv2d(out_channels, kernel_size=3, padding=1))
        layers.append(nn.ReLU())
    layers.append(nn.MaxPool2d(kernel_size=2, stride=2))
    return nn.Sequential(*layers)
```

In [ ]:
```
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)
```

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

```
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])
```

In [ ]:
```
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)
```



8.6 Residual Networks (ResNet) and ResNeXt

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

In [ ]:
```
class Residual(nn.Module):
    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)
```

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

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

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

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

In [ ]:
```
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))
```

In [ ]:
```
@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)
```

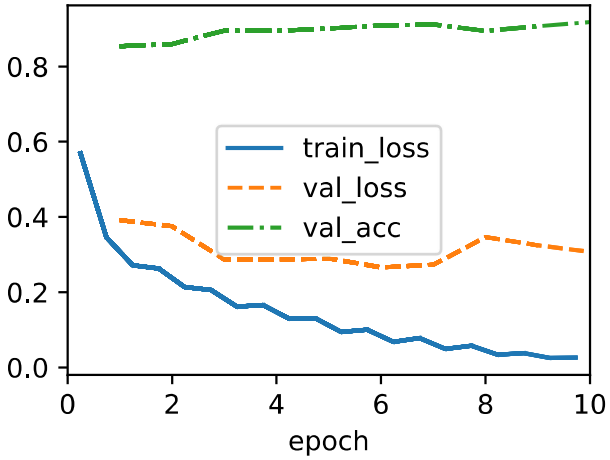
In [ ]:
```
@d2l.add_to_class(ResNet)
def __init__(self, arch, lr=0.1, num_classes=10):
    super(ResNet, self).__init__()
    self.save_hyperparameters()
    self.net = nn.Sequential(self.b1())
    for i, b in enumerate(arch):
        self.net.add_module(f'b{i+2}', self.block(*b, first_block=(i==0)))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

In [ ]:
```
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

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])
```

In [ ]:
```
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 [ ]:
```
class ResNextBlock(nn.Module):
    def __init__(self, num_channels, groups, bot_mul, use_1x1conv=False,
                 strides=1):
        super().__init__()
        bot_channels = int(round(num_channels * bot_mul))
        self.conv1 = nn.LazyConv2d(bot_channels, kernel_size=1, stride=1)
        self.conv2 = nn.LazyConv2d(bot_channels, kernel_size=3,
                                   stride=strides, padding=1,
                                   groups=bot_channels//groups)
        self.conv3 = nn.LazyConv2d(num_channels, kernel_size=1, stride=1)
        self.bn1 = nn.LazyBatchNorm2d()
        self.bn2 = nn.LazyBatchNorm2d()
        self.bn3 = nn.LazyBatchNorm2d()
        if use_1x1conv:
            self.conv4 = nn.LazyConv2d(num_channels, kernel_size=1,
                                       stride=strides)
            self.bn4 = nn.LazyBatchNorm2d()
        else:
            self.conv4 = None

    def forward(self, X):
        Y = F.relu(self.bn1(self.conv1(X)))
        Y = F.relu(self.bn2(self.conv2(X)))
        Y = self.bn3(self.conv3(Y))
        if self.conv4:
            X = self.bn4(self.conv4(X))
        return F.relu(Y + X)
```

In [ ]:
```
blk = ResNextBlock(32, 16, 1)
X = torch.randn(4, 32, 96, 96)
blk(X).shape
```

Out[ ]:
```
torch.Size([4, 32, 96, 96])
```

Discussion

7.1)

1. How much different between one-dimensional training model of ASR and two-dimensioanl training model of ASR. In CNN, can we use dimension of input data as hyperparameter?

7.2)

1. Optimized kernel tensor is remarkably close to the kernel tensor K we defined earlier -> Why?

2)Is there any difference at performance between convolution cross correlation?

--> In the process of training, there is optimization of filter so the result can be optimized.

7.3)

1. By determining size of the filter and strides, we can get different output data for the same size. Big size filter and bigger stride can make the same size of result of small size filter and small stride. It means that in the hidden layer, we can use stride and size of filter as hyperparameter as tuple. Then how can we chose the optimized hyperparameters for some case of training?

2. Zero-padding seems contaminting the input value of edge, then why zero-padding is widely used than mirror padding?

7.4)

1. Why 1x1 convolutional layer is widely used in CNN? 1x1 convolutional layer can not detect global or local features that has size bigger than 1x1 then why 1x1 convolutional layer is useful?

--> We can combine or shrinkg multi-channels and by using activation function 1x1 convolutional layer can add non-linearity to the model.

7.5)

1. Max-pooling seems that making output biased to the higher values and integrate values with max values. Then why is max-pooling is preferable than average-pooling?

--> Max-pooling has high performace at capturing important feature from the input data, therefore biased result can make strong feature more highlighted. Also, Average-pooling makes data blended not well-featured.

7.6)

1. Modern LeNet uses max-pooling an ReLU function, what is the difference between that kind of operation?

8.2)

1. VGG makes the number of feature map channels incrementally, so it seems that VGG make computational overhead too hardly then why is VGG prefered then the conventional CNN models?

--> It's the trade off between resource and the performace. VGG has higher computational cost but its image classification performace is quite good than prior models.

8.6)

1. Resnet has less overfitting problem and skip connection makes deep network more stable. Then why nowadays transformer model is widely used and what is the reason that transformer model outcomed?

--> Its the limitation of CNN that has difficulty to process global information.

In [ ]: