HW_2_

October 10, 2024

1 7.1. From Fully Connected Layers to Convolutions

1.0.1 Comparison of AlexNet and LeNet: Design Principles and Key Differences

The design concepts of AlexNet and LeNet are quite similar, but there are significant differences as well.

AlexNet is much deeper than the relatively smaller LeNet-5. AlexNet consists of eight layers: five convolutional layers, two fully connected hidden layers, and one fully connected output layer.

AlexNet uses ReLU as its activation function instead of sigmoid.

Dropout, ReLU, and preprocessing are other key steps to enhance performance in computer vision tasks.

2 7.2. Convolutions for Images

[1]:
```
!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)
  Using cached 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-cp38-cp38-win_amd64.whl.metadata (2.3 kB)
Collecting matplotlib==3.7.2 (from d2l==1.0.3)
  Downloading matplotlib-3.7.2-cp38-cp38-win_amd64.whl.metadata (5.8 kB)
Requirement already satisfied: matplotlib-inline==0.1.6 in
d:\anaconda3\envs\myenv\lib\site-packages (from d2l==1.0.3) (0.1.6)
Collecting requests==2.31.0 (from d2l==1.0.3)
  Using cached 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-cp38-cp38-win_amd64.whl.metadata (18 kB)
Collecting scipy==1.10.1 (from d2l==1.0.3)
  Downloading scipy-1.10.1-cp38-cp38-win_amd64.whl.metadata (58 kB)
Requirement already satisfied: notebook in d:\anaconda3\envs\myenv\lib\site-
packages (from jupyter==1.0.0->d2l==1.0.3) (7.2.2)
Collecting qtconsole (from jupyter==1.0.0->d2l==1.0.3)
  Downloading qtconsole-5.6.0-py3-none-any.whl.metadata (5.0 kB)
Collecting jupyter-console (from jupyter==1.0.0->d2l==1.0.3)
```

```
  Using cached jupyter_console-6.6.3-py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: nbconvert in d:\anaconda3\envs\myenv\lib\site-
packages (from jupyter==1.0.0->d2l==1.0.3) (7.16.4)
Requirement already satisfied: ipykernel in d:\anaconda3\envs\myenv\lib\site-
packages (from jupyter==1.0.0->d2l==1.0.3) (6.28.0)
Requirement already satisfied: ipywidgets in d:\anaconda3\envs\myenv\lib\site-
packages (from jupyter==1.0.0->d2l==1.0.3) (8.1.2)
Collecting contourpy>=1.0.1 (from matplotlib==3.7.2->d2l==1.0.3)
  Downloading contourpy-1.1.1-cp38-cp38-win_amd64.whl.metadata (5.9 kB)
Collecting cycler>=0.10 (from matplotlib==3.7.2->d2l==1.0.3)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib==3.7.2->d2l==1.0.3)
  Downloading fonttools-4.54.1-cp38-cp38-win_amd64.whl.metadata (167 kB)
Collecting kiwisolver>=1.0.1 (from matplotlib==3.7.2->d2l==1.0.3)
  Downloading kiwisolver-1.4.7-cp38-cp38-win_amd64.whl.metadata (6.4 kB)
Requirement already satisfied: packaging>=20.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from matplotlib==3.7.2->d2l==1.0.3)
(24.1)
Collecting pillow>=6.2.0 (from matplotlib==3.7.2->d2l==1.0.3)
  Downloading pillow-10.4.0-cp38-cp38-win_amd64.whl.metadata (9.3 kB)
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
d:\anaconda3\envs\myenv\lib\site-packages (from matplotlib==3.7.2->d2l==1.0.3)
(2.9.0.post0)
Requirement already satisfied: importlib-resources>=3.2.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from matplotlib==3.7.2->d2l==1.0.3)
(6.4.0)
Requirement already satisfied: traitlets in d:\anaconda3\envs\myenv\lib\site-
packages (from matplotlib-inline==0.1.6->d2l==1.0.3) (5.14.3)
Requirement already satisfied: pytz>=2020.1 in d:\anaconda3\envs\myenv\lib\site-
packages (from pandas==2.0.3->d2l==1.0.3) (2024.1)
Collecting tzdata>=2022.1 (from pandas==2.0.3->d2l==1.0.3)
  Downloading tzdata-2024.2-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in
d:\anaconda3\envs\myenv\lib\site-packages (from requests==2.31.0->d2l==1.0.3)
(3.3.2)
Requirement already satisfied: idna<4,>=2.5 in d:\anaconda3\envs\myenv\lib\site-
packages (from requests==2.31.0->d2l==1.0.3) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
d:\anaconda3\envs\myenv\lib\site-packages (from requests==2.31.0->d2l==1.0.3)
(2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
d:\anaconda3\envs\myenv\lib\site-packages (from requests==2.31.0->d2l==1.0.3)
(2024.8.30)
Requirement already satisfied: zipp>=3.1.0 in d:\anaconda3\envs\myenv\lib\site-
packages (from importlib-resources>=3.2.0->matplotlib==3.7.2->d2l==1.0.3)
(3.17.0)
```

```
Requirement already satisfied: six>=1.5 in d:\anaconda3\envs\myenv\lib\site-
packages (from python-dateutil>=2.7->matplotlib==3.7.2->d2l==1.0.3) (1.16.0)
Requirement already satisfied: comm>=0.1.1 in d:\anaconda3\envs\myenv\lib\site-
packages (from ipykernel->jupyter==1.0.0->d2l==1.0.3) (0.2.1)
Requirement already satisfied: debugpy>=1.6.5 in
d:\anaconda3\envs\myenv\lib\site-packages (from
ipykernel->jupyter==1.0.0->d2l==1.0.3) (1.6.7)
Requirement already satisfied: ipython>=7.23.1 in
d:\anaconda3\envs\myenv\lib\site-packages (from
ipykernel->jupyter==1.0.0->d2l==1.0.3) (8.12.2)
Requirement already satisfied: jupyter-client>=6.1.12 in
d:\anaconda3\envs\myenv\lib\site-packages (from
ipykernel->jupyter==1.0.0->d2l==1.0.3) (8.6.0)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in
d:\anaconda3\envs\myenv\lib\site-packages (from
ipykernel->jupyter==1.0.0->d2l==1.0.3) (5.7.2)
Requirement already satisfied: nest-asyncio in d:\anaconda3\envs\myenv\lib\site-
packages (from ipykernel->jupyter==1.0.0->d2l==1.0.3) (1.6.0)
Requirement already satisfied: psutil in d:\anaconda3\envs\myenv\lib\site-
packages (from ipykernel->jupyter==1.0.0->d2l==1.0.3) (5.9.0)
Requirement already satisfied: pyzmq>=24 in d:\anaconda3\envs\myenv\lib\site-
packages (from ipykernel->jupyter==1.0.0->d2l==1.0.3) (25.1.2)
Requirement already satisfied: tornado>=6.1 in d:\anaconda3\envs\myenv\lib\site-
packages (from ipykernel->jupyter==1.0.0->d2l==1.0.3) (6.4.1)
Requirement already satisfied: widgetsnbextension~=4.0.10 in
d:\anaconda3\envs\myenv\lib\site-packages (from
ipywidgets->jupyter==1.0.0->d2l==1.0.3) (4.0.10)
Requirement already satisfied: jupyterlab-widgets~=3.0.10 in
d:\anaconda3\envs\myenv\lib\site-packages (from
ipywidgets->jupyter==1.0.0->d2l==1.0.3) (3.0.10)
Requirement already satisfied: prompt-toolkit>=3.0.30 in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyter-
console->jupyter==1.0.0->d2l==1.0.3) (3.0.43)
Requirement already satisfied: pygments in d:\anaconda3\envs\myenv\lib\site-
packages (from jupyter-console->jupyter==1.0.0->d2l==1.0.3) (2.15.1)
Requirement already satisfied: beautifulsoup4 in
d:\anaconda3\envs\myenv\lib\site-packages (from
nbconvert->jupyter==1.0.0->d2l==1.0.3) (4.12.3)
Requirement already satisfied: bleach!=5.0.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from
nbconvert->jupyter==1.0.0->d2l==1.0.3) (4.1.0)
Requirement already satisfied: defusedxml in d:\anaconda3\envs\myenv\lib\site-
packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (0.7.1)
Requirement already satisfied: importlib-metadata>=3.6 in
d:\anaconda3\envs\myenv\lib\site-packages (from
nbconvert->jupyter==1.0.0->d2l==1.0.3) (7.0.1)
Requirement already satisfied: jinja2>=3.0 in d:\anaconda3\envs\myenv\lib\site-
packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (3.1.4)
```

```
Requirement already satisfied: jupyterlab-pygments in
d:\anaconda3\envs\myenv\lib\site-packages (from
nbconvert->jupyter==1.0.0->d2l==1.0.3) (0.1.2)
Requirement already satisfied: markupsafe>=2.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from
nbconvert->jupyter==1.0.0->d2l==1.0.3) (2.1.3)
Requirement already satisfied: mistune<4,>=2.0.3 in
d:\anaconda3\envs\myenv\lib\site-packages (from
nbconvert->jupyter==1.0.0->d2l==1.0.3) (2.0.4)
Requirement already satisfied: nbclient>=0.5.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from
nbconvert->jupyter==1.0.0->d2l==1.0.3) (0.8.0)
Requirement already satisfied: nbformat>=5.7 in
d:\anaconda3\envs\myenv\lib\site-packages (from
nbconvert->jupyter==1.0.0->d2l==1.0.3) (5.10.4)
Requirement already satisfied: pandocfilters>=1.4.1 in
d:\anaconda3\envs\myenv\lib\site-packages (from
nbconvert->jupyter==1.0.0->d2l==1.0.3) (1.5.0)
Requirement already satisfied: tinycss2 in d:\anaconda3\envs\myenv\lib\site-
packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (1.2.1)
Requirement already satisfied: jupyter-server<3,>=2.4.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from
notebook->jupyter==1.0.0->d2l==1.0.3) (2.14.1)
Requirement already satisfied: jupyterlab-server<3,>=2.27.1 in
d:\anaconda3\envs\myenv\lib\site-packages (from
notebook->jupyter==1.0.0->d2l==1.0.3) (2.27.3)
Requirement already satisfied: jupyterlab<4.3,>=4.2.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from
notebook->jupyter==1.0.0->d2l==1.0.3) (4.2.5)
Requirement already satisfied: notebook-shim<0.3,>=0.2 in
d:\anaconda3\envs\myenv\lib\site-packages (from
notebook->jupyter==1.0.0->d2l==1.0.3) (0.2.3)
Collecting qtpy>=2.4.0 (from qtconsole->jupyter==1.0.0->d2l==1.0.3)
  Using cached QtPy-2.4.1-py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: webencodings in d:\anaconda3\envs\myenv\lib\site-
packages (from bleach!=5.0.0->nbconvert->jupyter==1.0.0->d2l==1.0.3) (0.5.1)
Requirement already satisfied: backcall in d:\anaconda3\envs\myenv\lib\site-
packages (from ipython>=7.23.1->ipykernel->jupyter==1.0.0->d2l==1.0.3) (0.2.0)
Requirement already satisfied: decorator in d:\anaconda3\envs\myenv\lib\site-
packages (from ipython>=7.23.1->ipykernel->jupyter==1.0.0->d2l==1.0.3) (5.1.1)
Requirement already satisfied: jedi>=0.16 in d:\anaconda3\envs\myenv\lib\site-
packages (from ipython>=7.23.1->ipykernel->jupyter==1.0.0->d2l==1.0.3) (0.19.1)
Requirement already satisfied: pickleshare in d:\anaconda3\envs\myenv\lib\site-
packages (from ipython>=7.23.1->ipykernel->jupyter==1.0.0->d2l==1.0.3) (0.7.5)
Requirement already satisfied: stack-data in d:\anaconda3\envs\myenv\lib\site-
packages (from ipython>=7.23.1->ipykernel->jupyter==1.0.0->d2l==1.0.3) (0.2.0)
Requirement already satisfied: typing-extensions in
d:\anaconda3\envs\myenv\lib\site-packages (from
```

```
ipython>=7.23.1->ipykernel->jupyter==1.0.0->d2l==1.0.3) (4.11.0)
Requirement already satisfied: colorama in d:\anaconda3\envs\myenv\lib\site-
packages (from ipython>=7.23.1->ipykernel->jupyter==1.0.0->d2l==1.0.3) (0.4.6)
Requirement already satisfied: platformdirs>=2.5 in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyter-
core!=5.0.*,>=4.12->ipykernel->jupyter==1.0.0->d2l==1.0.3) (3.10.0)
Requirement already satisfied: pywin32>=300 in d:\anaconda3\envs\myenv\lib\site-
packages (from jupyter-
core!=5.0.*,>=4.12->ipykernel->jupyter==1.0.0->d2l==1.0.3) (305.1)
Requirement already satisfied: anyio>=3.1.0 in d:\anaconda3\envs\myenv\lib\site-
packages (from jupyter-server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3)
(4.2.0)
Requirement already satisfied: argon2-cffi>=21.1 in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (21.3.0)
Requirement already satisfied: jupyter-events>=0.9.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (0.10.0)
Requirement already satisfied: jupyter-server-terminals>=0.4.4 in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (0.4.4)
Requirement already satisfied: overrides>=5.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (7.4.0)
Requirement already satisfied: prometheus-client>=0.9 in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (0.14.1)
Requirement already satisfied: pywinpty>=2.0.1 in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (2.0.10)
Requirement already satisfied: send2trash>=1.8.2 in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (1.8.2)
Requirement already satisfied: terminado>=0.8.3 in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (0.17.1)
Requirement already satisfied: websocket-client>=1.7 in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (1.8.0)
Requirement already satisfied: async-lru>=1.0.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from
jupyterlab<4.3,>=4.2.0->notebook->jupyter==1.0.0->d2l==1.0.3) (2.0.4)
Requirement already satisfied: httpx>=0.25.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from
jupyterlab<4.3,>=4.2.0->notebook->jupyter==1.0.0->d2l==1.0.3) (0.27.0)
Requirement already satisfied: jupyter-lsp>=2.0.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from
jupyterlab<4.3,>=4.2.0->notebook->jupyter==1.0.0->d2l==1.0.3) (2.2.0)
```

```
Requirement already satisfied: setuptools>=40.1.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from
jupyterlab<4.3,>=4.2.0->notebook->jupyter==1.0.0->d2l==1.0.3) (75.1.0)
Requirement already satisfied: tomli>=1.2.2 in d:\anaconda3\envs\myenv\lib\site-
packages (from jupyterlab<4.3,>=4.2.0->notebook->jupyter==1.0.0->d2l==1.0.3)
(2.0.1)
Requirement already satisfied: babel>=2.10 in d:\anaconda3\envs\myenv\lib\site-
packages (from jupyterlab-
server<3,>=2.27.1->notebook->jupyter==1.0.0->d2l==1.0.3) (2.11.0)
Requirement already satisfied: json5>=0.9.0 in d:\anaconda3\envs\myenv\lib\site-
packages (from jupyterlab-
server<3,>=2.27.1->notebook->jupyter==1.0.0->d2l==1.0.3) (0.9.6)
Requirement already satisfied: jsonschema>=4.18.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyterlab-
server<3,>=2.27.1->notebook->jupyter==1.0.0->d2l==1.0.3) (4.23.0)
Requirement already satisfied: fastjsonschema>=2.15 in
d:\anaconda3\envs\myenv\lib\site-packages (from
nbformat>=5.7->nbconvert->jupyter==1.0.0->d2l==1.0.3) (2.16.2)
Requirement already satisfied: wcwidth in d:\anaconda3\envs\myenv\lib\site-
packages (from prompt-toolkit>=3.0.30->jupyter-
console->jupyter==1.0.0->d2l==1.0.3) (0.2.5)
Requirement already satisfied: soupsieve>1.2 in
d:\anaconda3\envs\myenv\lib\site-packages (from
beautifulsoup4->nbconvert->jupyter==1.0.0->d2l==1.0.3) (2.5)
Requirement already satisfied: sniffio>=1.1 in d:\anaconda3\envs\myenv\lib\site-
packages (from anyio>=3.1.0->jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (1.3.0)
Requirement already satisfied: exceptiongroup>=1.0.2 in
d:\anaconda3\envs\myenv\lib\site-packages (from anyio>=3.1.0->jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (1.2.0)
Requirement already satisfied: argon2-cffi-bindings in
d:\anaconda3\envs\myenv\lib\site-packages (from argon2-cffi>=21.1->jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (21.2.0)
Requirement already satisfied: httpcore==1.* in
d:\anaconda3\envs\myenv\lib\site-packages (from
httpx>=0.25.0->jupyterlab<4.3,>=4.2.0->notebook->jupyter==1.0.0->d2l==1.0.3)
(1.0.2)
Requirement already satisfied: h11<0.15,>=0.13 in
d:\anaconda3\envs\myenv\lib\site-packages (from httpcore==1.*->httpx>=0.25.0->ju
pyterlab<4.3,>=4.2.0->notebook->jupyter==1.0.0->d2l==1.0.3) (0.14.0)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in
d:\anaconda3\envs\myenv\lib\site-packages (from
jedi>=0.16->ipython>=7.23.1->ipykernel->jupyter==1.0.0->d2l==1.0.3) (0.8.3)
Requirement already satisfied: attrs>=22.2.0 in
d:\anaconda3\envs\myenv\lib\site-packages (from jsonschema>=4.18.0->jupyterlab-
server<3,>=2.27.1->notebook->jupyter==1.0.0->d2l==1.0.3) (23.1.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
d:\anaconda3\envs\myenv\lib\site-packages (from jsonschema>=4.18.0->jupyterlab-
```

```
server<3,>=2.27.1->notebook->jupyter==1.0.0->d2l==1.0.3) (2023.7.1)
Requirement already satisfied: pkgutil-resolve-name>=1.3.10 in
d:\anaconda3\envs\myenv\lib\site-packages (from jsonschema>=4.18.0->jupyterlab-
server<3,>=2.27.1->notebook->jupyter==1.0.0->d2l==1.0.3) (1.3.10)
Requirement already satisfied: referencing>=0.28.4 in
d:\anaconda3\envs\myenv\lib\site-packages (from jsonschema>=4.18.0->jupyterlab-
server<3,>=2.27.1->notebook->jupyter==1.0.0->d2l==1.0.3) (0.30.2)
Requirement already satisfied: rpds-py>=0.7.1 in
d:\anaconda3\envs\myenv\lib\site-packages (from jsonschema>=4.18.0->jupyterlab-
server<3,>=2.27.1->notebook->jupyter==1.0.0->d2l==1.0.3) (0.10.6)
Requirement already satisfied: python-json-logger>=2.0.4 in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyter-events>=0.9.0->jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (2.0.7)
Requirement already satisfied: pyyaml>=5.3 in d:\anaconda3\envs\myenv\lib\site-
packages (from jupyter-events>=0.9.0->jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (6.0.1)
Requirement already satisfied: rfc3339-validator in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyter-events>=0.9.0->jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (0.1.4)
Requirement already satisfied: rfc3986-validator>=0.1.1 in
d:\anaconda3\envs\myenv\lib\site-packages (from jupyter-events>=0.9.0->jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (0.1.1)
Requirement already satisfied: executing in d:\anaconda3\envs\myenv\lib\site-
packages (from stack-
data->ipython>=7.23.1->ipykernel->jupyter==1.0.0->d2l==1.0.3) (0.8.3)
Requirement already satisfied: asttokens in d:\anaconda3\envs\myenv\lib\site-
packages (from stack-
data->ipython>=7.23.1->ipykernel->jupyter==1.0.0->d2l==1.0.3) (2.0.5)
Requirement already satisfied: pure-eval in d:\anaconda3\envs\myenv\lib\site-
packages (from stack-
data->ipython>=7.23.1->ipykernel->jupyter==1.0.0->d2l==1.0.3) (0.2.2)
Collecting fqdn (from jsonschema[format-nongpl]>=4.18.0->jupyter-
events>=0.9.0->jupyter-server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3)
  Downloading fqdn-1.5.1-py3-none-any.whl.metadata (1.4 kB)
Collecting isoduration (from jsonschema[format-nongpl]>=4.18.0->jupyter-
events>=0.9.0->jupyter-server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3)
  Downloading isoduration-20.11.0-py3-none-any.whl.metadata (5.7 kB)
Collecting jsonpointer>1.13 (from jsonschema[format-nongpl]>=4.18.0->jupyter-
events>=0.9.0->jupyter-server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3)
  Downloading jsonpointer-3.0.0-py2.py3-none-any.whl.metadata (2.3 kB)
Collecting uri-template (from jsonschema[format-nongpl]>=4.18.0->jupyter-
events>=0.9.0->jupyter-server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3)
  Downloading uri_template-1.3.0-py3-none-any.whl.metadata (8.8 kB)
Collecting webcolors>=24.6.0 (from jsonschema[format-nongpl]>=4.18.0->jupyter-
events>=0.9.0->jupyter-server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3)
  Downloading webcolors-24.8.0-py3-none-any.whl.metadata (2.6 kB)
Requirement already satisfied: cffi>=1.0.1 in d:\anaconda3\envs\myenv\lib\site-
packages (from argon2-cffi-bindings->argon2-cffi>=21.1->jupyter-
```

```
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (1.17.1)
Requirement already satisfied: pycparser in d:\anaconda3\envs\myenv\lib\site-
packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi>=21.1->jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3) (2.21)
Collecting arrow>=0.15.0 (from isoduration->jsonschema[format-
nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-
server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3)
  Downloading arrow-1.3.0-py3-none-any.whl.metadata (7.5 kB)
Collecting types-python-dateutil>=2.8.10 (from
arrow>=0.15.0->isoduration->jsonschema[format-nongpl]>=4.18.0->jupyter-
events>=0.9.0->jupyter-server<3,>=2.4.0->notebook->jupyter==1.0.0->d2l==1.0.3)
  Downloading types_python_dateutil-2.9.0.20241003-py3-none-any.whl.metadata
(1.9 kB)
Downloading d2l-1.0.3-py3-none-any.whl (111 kB)
Using cached jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Downloading matplotlib-3.7.2-cp38-cp38-win_amd64.whl (7.5 MB)
   ---------------------------------------- 0.0/7.5 MB ? eta -:--:--
   ----------- ---------------------------- 2.4/7.5 MB 12.3 MB/s eta 0:00:01
   ------------------------ -------------- 4.7/7.5 MB 11.9 MB/s eta 0:00:01
   ---------------------------------------  7.3/7.5 MB 11.9 MB/s eta 0:00:01
   ---------------------------------------- 7.5/7.5 MB 11.6 MB/s eta 0:00:00
Downloading numpy-1.23.5-cp38-cp38-win_amd64.whl (14.7 MB)
   ---------------------------------------- 0.0/14.7 MB ? eta -:--:--
   ------ --------------------------------- 2.4/14.7 MB 12.2 MB/s eta 0:00:02
   ------------ --------------------------- 5.0/14.7 MB 12.1 MB/s eta 0:00:01
   ------------------- -------------------- 7.3/14.7 MB 11.9 MB/s eta 0:00:01
   ------------------------- -------------- 9.7/14.7 MB 11.8 MB/s eta 0:00:01
   ------------------------------- -------- 12.3/14.7 MB 11.9 MB/s eta 0:00:01
   ---------------------------------------  14.4/14.7 MB 11.9 MB/s eta 0:00:01
   ---------------------------------------- 14.7/14.7 MB 11.7 MB/s eta 0:00:00
Downloading pandas-2.0.3-cp38-cp38-win_amd64.whl (10.8 MB)
   ---------------------------------------- 0.0/10.8 MB ? eta -:--:--
   -------- ------------------------------- 2.4/10.8 MB 11.2 MB/s eta 0:00:01
   ----------------- ---------------------- 5.0/10.8 MB 11.6 MB/s eta 0:00:01
   ------------------------- -------------- 7.3/10.8 MB 11.6 MB/s eta 0:00:01
   ---------------------------------- ----- 9.7/10.8 MB 11.8 MB/s eta 0:00:01
   ---------------------------------------- 10.8/10.8 MB 11.6 MB/s eta 0:00:00
Using cached requests-2.31.0-py3-none-any.whl (62 kB)
Downloading scipy-1.10.1-cp38-cp38-win_amd64.whl (42.2 MB)
   ---------------------------------------- 0.0/42.2 MB ? eta -:--:--
   -- ------------------------------------- 2.4/42.2 MB 12.3 MB/s eta 0:00:04
   ---- ----------------------------------- 5.0/42.2 MB 12.1 MB/s eta 0:00:04
   ------ --------------------------------- 7.3/42.2 MB 11.9 MB/s eta 0:00:03
   --------- ------------------------------ 9.7/42.2 MB 11.9 MB/s eta 0:00:03
   ----------- ---------------------------- 12.3/42.2 MB 11.9 MB/s eta 0:00:03
   ------------- -------------------------- 14.7/42.2 MB 11.8 MB/s eta 0:00:03
   ---------------- ----------------------- 17.3/42.2 MB 11.9 MB/s eta 0:00:03
   ------------------ --------------------- 19.7/42.2 MB 11.8 MB/s eta 0:00:02
```

```
   ---------------------- ------------------ 22.3/42.2 MB 11.8 MB/s eta 0:00:02
   ------------------------ ---------------- 24.6/42.2 MB 11.8 MB/s eta 0:00:02
   -------------------------- -------------- 27.3/42.2 MB 11.8 MB/s eta 0:00:02
   ---------------------------- ------------ 29.6/42.2 MB 11.8 MB/s eta 0:00:02
   ------------------------------ ---------- 32.0/42.2 MB 11.8 MB/s eta 0:00:01
   -------------------------------- ------- 34.6/42.2 MB 11.8 MB/s eta 0:00:01
   ---------------------------------- ----- 37.0/42.2 MB 11.8 MB/s eta 0:00:01
   ------------------------------------ --- 39.6/42.2 MB 11.8 MB/s eta 0:00:01
   --------------------------------------  41.9/42.2 MB 11.8 MB/s eta 0:00:01
   ---------------------------------------- 42.2/42.2 MB 11.7 MB/s eta 0:00:00
Downloading contourpy-1.1.1-cp38-cp38-win_amd64.whl (477 kB)
Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.54.1-cp38-cp38-win_amd64.whl (1.5 MB)
   ---------------------------------------- 0.0/1.5 MB ? eta -:--:--
   ---------------------------------------- 1.5/1.5 MB 11.7 MB/s eta 0:00:00
Downloading kiwisolver-1.4.7-cp38-cp38-win_amd64.whl (55 kB)
Downloading pillow-10.4.0-cp38-cp38-win_amd64.whl (2.6 MB)
   ---------------------------------------- 0.0/2.6 MB ? eta -:--:--
   ---------------------------------- ----- 2.4/2.6 MB 11.2 MB/s eta 0:00:01
   ---------------------------------------- 2.6/2.6 MB 11.3 MB/s eta 0:00:00
Downloading pyparsing-3.0.9-py3-none-any.whl (98 kB)
Downloading tzdata-2024.2-py2.py3-none-any.whl (346 kB)
Using cached jupyter_console-6.6.3-py3-none-any.whl (24 kB)
Downloading qtconsole-5.6.0-py3-none-any.whl (124 kB)
Using cached QtPy-2.4.1-py3-none-any.whl (93 kB)
Downloading jsonpointer-3.0.0-py2.py3-none-any.whl (7.6 kB)
Downloading webcolors-24.8.0-py3-none-any.whl (15 kB)
Downloading fqdn-1.5.1-py3-none-any.whl (9.1 kB)
Downloading isoduration-20.11.0-py3-none-any.whl (11 kB)
Downloading uri_template-1.3.0-py3-none-any.whl (11 kB)
Downloading arrow-1.3.0-py3-none-any.whl (66 kB)
Downloading types_python_dateutil-2.9.0.20241003-py3-none-any.whl (9.7 kB)
Installing collected packages: webcolors, uri-template, tzdata, types-python-
dateutil, requests, qtpy, pyparsing, pillow, numpy, kiwisolver, jsonpointer,
fqdn, fonttools, cycler, scipy, pandas, contourpy, arrow, matplotlib,
isoduration, qtconsole, jupyter-console, jupyter, d2l
  Attempting uninstall: requests
    Found existing installation: requests 2.32.3
    Uninstalling requests-2.32.3:
      Successfully uninstalled requests-2.32.3
Successfully installed arrow-1.3.0 contourpy-1.1.1 cycler-0.12.1 d2l-1.0.3
fonttools-4.54.1 fqdn-1.5.1 isoduration-20.11.0 jsonpointer-3.0.0 jupyter-1.0.0
jupyter-console-6.6.3 kiwisolver-1.4.7 matplotlib-3.7.2 numpy-1.23.5
pandas-2.0.3 pillow-10.4.0 pyparsing-3.0.9 qtconsole-5.6.0 qtpy-2.4.1
requests-2.31.0 scipy-1.10.1 types-python-dateutil-2.9.0.20241003 tzdata-2024.2
uri-template-1.3.0 webcolors-24.8.0
```

2.0.1 Convolutions for Images

[1]:
```
import torch
from torch import nn
from d2l import torch as d2l
```

2.0.2 The Cross-Correlation Operation

[2]:
```
def corr2d(X, K):
    """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
```

[3]:
```
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)
```

[3]:
```
tensor([[19., 25.],
        [37., 43.]])
```

2.0.3 Convolutional Layers

[4]:
```
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
```

2.0.4 Object Edge Detection in Images

[5]:
```
X = torch.ones((6, 8))
X[:, 2:6] = 0
X
```

[5]:
```
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.]])
```

[6]:
```
K = torch.tensor([[1.0, -1.0]])
```

[7]:
```
Y = corr2d(X, K)
Y
```

[7]:
```
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.]])
```

[8]:
```
corr2d(X.t(), K)
```

[8]:
```
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.]])
```

2.0.5 Learning a Kernel

[9]:
```
# Construct a two-dimensional convolutional layer with 1 output channel and a
# kernel of shape (1, 2). For the sake of simplicity, we ignore the bias here
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 batch
# size (number of examples in the batch) and the number of channels are both 1
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 9.821
epoch 4, loss 3.126
```

```
epoch 6, loss 1.130
epoch 8, loss 0.438
epoch 10, loss 0.175
```

[10]:
```
conv2d.weight.data.reshape((1, 2))
```

[10]:
```
tensor([[ 0.9471, -1.0328]])
```

2.0.6 Summary

The core computation required for a convolutional layer is a cross-correlation operation. As can be seen, the computation is straightforward and, most importantly, highly local. This affords significant hardware optimization and many recent results in computer vision are only possible because of that. After all, it means that chip designers can invest in fast computation rather than memory when it comes to optimizing for convolutions. While this may not lead to optimal designs for other applications, it does open the door to ubiquitous and affordable computer vision.

3 7.3. Padding and Stride

[11]:
```
import torch
from torch import nn
```

3.0.1 Padding

[12]:
```
# We define a helper function to calculate convolutions. It initializes the
# 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 1
    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 columns
# are added
conv2d = nn.LazyConv2d(1, kernel_size=3, padding=1)
X = torch.rand(size=(8, 8))
comp_conv2d(conv2d, X).shape
```

[12]:
```
torch.Size([8, 8])
```

[13]:
```
# We use a convolution kernel with height 5 and width 3. The padding on either
# 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
```

[13]:
```
torch.Size([8, 8])
```

3.0.2 Stride

[14]:
```
conv2d = nn.LazyConv2d(1, kernel_size=3, padding=1, stride=2)
comp_conv2d(conv2d, X).shape
```

[14]:
```
torch.Size([4, 4])
```

[15]:
```
conv2d = nn.LazyConv2d(1, kernel_size=(3, 5), padding=(0, 1), stride=(3, 4))
comp_conv2d(conv2d, X).shape
```

[15]:
```
torch.Size([2, 2])
```

3.0.3 Summary and Discussion

Padding can increase the height and width of the output. This is often used to give the output the same height and width as the input to avoid undesirable shrinkage of the output. Moreover, it ensures that all pixels are used equally frequently.

The stride can reduce the resolution of the output, for example reducing the height and width of the output to only 1/n of the height and width of the input for n>1 . By default, the padding is 0 and the stride is 1.

Moreover, operators can be engineered to take advantage of this padding implicitly without the need to allocate additional memory. At the same time, it allows CNNs to encode implicit position information within an image, simply by learning where the “whitespace” is. There are many alternatives to zero-padding. :citet:Alsallakh.Kokhlikyan.Miglani.ea.2020 provided an extensive overview of those (albeit without a clear case for when to use nonzero paddings unless artifacts occur).

4 7.4. Multiple Input and Multiple Output Channels

[16]:
```
import torch
from d2l import torch as d2l
```

4.0.1 Multiple Input Channels

[17]:
```
  def corr2d_multi_in(X, K):
      # Iterate through the 0th dimension (channel) of K first, then add them up
      return sum(d2l.corr2d(x, k) for x, k in zip(X, K))
```

[18]:
```
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)
```

[18]:
```
tensor([[ 56.,  72.],
        [104., 120.]])
```

4.0.2 Multiple Output Channels

[19]:
```
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)
```

[20]:
```
K = torch.stack((K, K + 1, K + 2), 0)
K.shape
```

[20]:
```
torch.Size([3, 2, 2, 2])
```

[21]:
```
corr2d_multi_in_out(X, K)
```

[21]:
```
tensor([[[ 56.,  72.],
         [104., 120.]],

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

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

4.0.3 1×1 Convolutional Layer

[22]:
```
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))
```

[23]:
```
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
```

4.0.4 Discussion

Channels allow us to combine the best of both worlds: MLPs that allow for significant nonlinearities and convolutions that allow for localized analysis of features. In particular, channels allow the CNN

to reason with multiple features, such as edge and shape detectors at the same time. They also offer a practical trade-off between the drastic parameter reduction arising from translation invariance and locality, and the need for expressive and diverse models in computer vision.

5 7.5. Pooling

[24]:
```
import torch
from torch import nn
from d2l import torch as d2l
```

5.0.1 Maximum Pooling and Average Pooling

[25]:
```
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
```

[26]:
```
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))
```

[26]:
```
tensor([[4., 5.],
        [7., 8.]])
```

[27]:
```
pool2d(X, (2, 2), 'avg')
```

[27]:
```
tensor([[2., 3.],
        [5., 6.]])
```

5.0.2 [Padding and Stride]

[28]:
```
X = torch.arange(16, dtype=torch.float32).reshape((1, 1, 4, 4))
X
```

[28]:
```
tensor([[[[ 0.,  1.,  2.,  3.],
          [ 4.,  5.,  6.,  7.],
          [ 8.,  9., 10., 11.],
          [12., 13., 14., 15.]]]])
```

[29]:
```
pool2d = nn.MaxPool2d(3, padding=1, stride=2)
pool2d(X)
```

[29]:
```
tensor([[[[ 5.,  7.],
          [13., 15.]]]])
```

[30]:
```
pool2d = nn.MaxPool2d((2, 3), stride=(2, 3), padding=(0, 1))
pool2d(X)
```

[30]:
```
tensor([[[[ 5.,  7.],
          [13., 15.]]]])
```

5.0.3 Multiple Channels

[31]:
```
X = torch.cat((X, X + 1), 1)
X
```

[31]:
```
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.]]]])
```

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

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

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

5.0.4 Summary

Pooling is an exceedingly simple operation. It does exactly what its name indicates, aggregate results over a window of values. All convolution semantics, such as strides and padding apply in the same way as they did previously. Note that pooling is indifferent to channels, i.e., it leaves the number of channels unchanged and it applies to each channel separately. Lastly, of the two popular pooling choices, max-pooling is preferable to average pooling, as it confers some degree of invariance to output. A popular choice is to pick a pooling window size of 2×2 to quarter the spatial resolution of output.

6 7.6. Convolutional Neural Networks (LeNet)

[33]:
```
import torch
from torch import nn
from d2l import torch as d2l
```

6.0.1 LeNet

[34]:
```
def init_cnn(module):
    """Initialize weights for CNNs."""
    if type(module) == nn.Linear or type(module) == nn.Conv2d:
        nn.init.xavier_uniform_(module.weight)
```

[35]:
```
class LeNet(d2l.Classifier):
    """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))
```

[36]:
```
@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])
```

6.0.2 Summary

The architectures proposed, e.g., in the form of LeNet-5 remain meaningful, even to this day. It is worth comparing the error rates on Fashion-MNIST achievable with LeNet-5 both to the very best possible with MLPs (:numref:sec_mlp-implementation) and those with significantly more advanced architectures such as ResNet (:numref:sec_resnet). LeNet is much more similar to the latter than to the former. One of the primary differences, as we shall see, is that greater amounts of computation enabled significantly more complex architectures.

7 8.2. Networks Using Blocks (VGG)

[38]:
```
import torch
from torch import nn
from d2l import torch as d2l
```

7.0.1 (VGG Blocks)

[39]:
```
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)
```

7.0.2 [VGG Network]

[40]:
```
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)
```

[41]:
```
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])
```

7.0.3 Summary

One might argue that VGG is the first truly modern convolutional neural network. While AlexNet introduced many of the components of what make deep learning effective at scale, it is VGG that arguably introduced key properties such as blocks of multiple convolutions and a preference for deep and narrow networks. It is also the first network that is actually an entire family of similarly parametrized models, giving the practitioner ample trade-off between complexity and speed. This is also the place where modern deep learning frameworks shine. It is no longer necessary to generate XML configuration files to specify a network but rather, to assemble said networks through simple Python code.

8 8.6. Residual Networks (ResNet) and ResNeXt

[43]:
```
import torch
from torch import nn
from torch.nn import functional as F
from d2l import torch as d2l
```

[44]:
```
class Residual(nn.Module):
    """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)
```

[45]:
```
blk = Residual(3)
X = torch.randn(4, 3, 6, 6)
blk(X).shape
```

[45]:
```
torch.Size([4, 3, 6, 6])
```

[46]:
```
blk = Residual(6, use_1x1conv=True, strides=2)
blk(X).shape
```

[46]:
```
torch.Size([4, 6, 3, 3])
```

8.0.1 [ResNet Model]

[47]:
```
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))
```

[48]:
```
@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)
```

[49]:
```
@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)
```

[50]:
```
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)
```

[51]:
```
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])
```