COSE474-2024F : Deep Learning HW1

Lab Session1

0.1 Installation

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_64.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.manylinux
2014_x86_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_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 2.7 MB/s eta 0:
00:00
Requirement already satisfied: notebook in /usr/local/lib/python3.10/dist-
packages (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/dis
t-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.
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/d
ist-packages (from matplotlib==3.7.2->d2l==1.0.3) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python
```

```
3.10/dist-packages (from matplotlib==3.7.2->d2l==1.0.3) (4.53.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python
3.10/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/
dist-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/pyth
on3.10/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/d
ist-packages (from pandas==2.0.3->d2l==1.0.3) (2024.2)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.1
0/dist-packages (from pandas==2.0.3->d2l==1.0.3) (2024.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/
python3.10/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/d
ist-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) (2.0.7)
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-
packages (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/python3.1
0/dist-packages (from ipykernel->jupyter==1.0.0->d2l==1.0.3) (7.34.0)
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.1
0/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/d
ist-packages (from ipykernel->jupyter==1.0.0->d2l==1.0.3) (6.3.3)
Requirement already satisfied: widgetsnbextension~=3.6.0 in /usr/local/li
b/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/li
b/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->jupyte
r==1.0.0->d2l==1.0.3) (3.0.47)
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-
packages (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-pack
ages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (4.9.4)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.1
0/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-pa
ckages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (6.1.0)
```

```
Requirement already satisfied: defusedxml in /usr/local/lib/python3.10/dis
t-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/di
st-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/python
3.10/dist-packages (from nbconvert->jupyter==1.0.0->d2l==1.0.3) (5.7.2)
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/pytho
n3.10/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/python
3.10/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/
dist-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/pyth
on3.10/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-
packages (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-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/di
st-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/python
3.10/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/python
3.10/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.
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/python
3.10/dist-packages (from notebook->jupyter==1.0.0->d2l==1.0.3) (0.20.0)
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
-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/di
st-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-
packages (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/di
st-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/python
3.10/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/pyth
on3.10/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/pyth
on3.10/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-p
ackages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->jupyter-consol
e->jupyter==1.0.0->d2l==1.0.3) (0.2.13)
Requirement already satisfied: ptyprocess in /usr/local/lib/python3.10/dis
t-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/pyth
on3.10/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-packages (from beautifulsoup4->nbconvert->jupyter==1.0.0->d2l==1.0.3)
(2.6)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/d
ist-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/pytho
n3.10/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/
dist-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/pytho
n3.10/dist-packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyt
er==1.0.0->d2l==1.0.3) (0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.1
0/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/py
thon3.10/dist-packages (from notebook-shim>=0.2.3->nbclassic>=0.4.7->noteb
ook->jupyter==1.0.0->d2l==1.0.3) (1.24.0)
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.10/di
st-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->
jupyter==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->nbclas
sic>=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.
10/dist-packages (from jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbcla
ssic>=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/d
ist-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/python3.1
0/dist-packages (from anyio<4,>=3.1.0->jupyter-server<3,>=1.8->notebook-sh
im>=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 5.5 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.manylinux20
14_x86_64.whl (11.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.6/11.6 MB 68.6 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_x
86_64.whl (17.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.1/17.1 MB 46.5 MB/s eta 0:0
0:00
Downloading pandas-2.0.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x
86_64.whl (12.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.3/12.3 MB 52.1 MB/s eta 0:0
0:00
Downloading requests-2.31.0-py3-none-any.whl (62 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.6/62.6 kB 3.2 MB/s eta 0:0
0:00
Downloading scipy-1.10.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x
86_64.whl (34.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 34.4/34.4 MB 12.4 MB/s eta 0:0
0:00
Downloading pyparsing-3.0.9-py3-none-any.whl (98 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.3/98.3 kB 4.4 MB/s eta 0:0
0:00
Downloading qtconsole-5.6.0-py3-none-any.whl (124 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 124.7/124.7 kB 7.1 MB/s eta 0:
00:00
Downloading QtPy-2.4.1-py3-none-any.whl (93 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 93.5/93.5 kB 6.4 MB/s eta 0:0
0:00
Using cached jedi-0.19.1-py2.py3-none-any.whl (1.6 MB)
Installing collected packages: requests, qtpy, pyparsing, numpy, matplotli
b-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 2.1.4
    Uninstalling pandas-2.1.4:
      Successfully uninstalled pandas-2.1.4
  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 packages that are installed. This behaviour is the source of the follo
wing dependency conflicts.
albucore 0.0.16 requires numpy>=1.24, but you have numpy 1.23.5 which is i
ncompatible.
albumentations 1.4.15 requires numpy>=1.24.4, but you have numpy 1.23.5 wh
ich is incompatible.
bigframes 1.17.0 requires numpy>=1.24.0, but you have numpy 1.23.5 which i
s incompatible.
chex 0.1.86 requires numpy>=1.24.1, but you have numpy 1.23.5 which is inc
ompatible.
google-colab 1.0.0 requires pandas==2.1.4, 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.
mizani 0.11.4 requires pandas>=2.1.0, but you have pandas 2.0.3 which is i
ncompatible.
pandas-stubs 2.1.4.231227 requires numpy>=1.26.0; python_version < "3.13",
but you have numpy 1.23.5 which is incompatible.
plotnine 0.13.6 requires pandas<3.0.0,>=2.1.0, but you have pandas 2.0.3 w
hich is incompatible.
xarray 2024.9.0 requires numpy>=1.24, but you have numpy 1.23.5 which is i
ncompatible.
xarray 2024.9.0 requires pandas>=2.1, but you have pandas 2.0.3 which is i
ncompatible.
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 qtcons
ole-5.6.0 qtpy-2.4.1 requests-2.31.0 scipy-1.10.1
```

2.1 Data manipulation

2.1.1 Getting Started

```
In [ ]: import torch
```

```
In [ ]: x=torch.arange(12,dtype=torch.float32)
        x
```

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

```
In [ ]: x.numel()
```

```
Out[ ]: 12
```

```
In [ ]: x.shape
```

```
Out[ ]: torch.Size([12])
```

```
In [ ]: X=x.reshape(3,4)
```

```
In [ ]: X
```

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

```
In [ ]: X.shape
```

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

```
In [ ]: torch.zeros(2,3,4)
```

```
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.]]])
```

```
In [ ]: torch.ones(2,3,4)
```

```
Out[ ]: tensor([[[1., 1., 1., 1.],
                 [1., 1., 1., 1.],
                 [1., 1., 1., 1.]],

                [[1., 1., 1., 1.],
                 [1., 1., 1., 1.],
                 [1., 1., 1., 1.]]])
```

```
In [ ]: torch.randn(3,4)
```

```
Out[ ]: tensor([[ 0.3332, -0.6325, -0.3552,  0.0752],
                [-1.3111,  0.4162, -0.0747,  0.3223],
                [ 0.7357,  0.8918, -0.2410, -0.5157]])
```

```
In [ ]: torch.tensor([[2, 1, 4, 3], [1, 2, 3, 4], [4, 3, 2, 1]])
```

```
Out[ ]: tensor([[2, 1, 4, 3],
                [1, 2, 3, 4],
                [4, 3, 2, 1]])
```

2.1.2. Indexing and Slicing

```
In [ ]: X[-1], X[1:3]
```

```
Out[ ]: (tensor([ 8.,  9., 10., 11.]),
         tensor([[ 4.,  5.,  6.,  7.],
                 [ 8.,  9., 10., 11.]]))
```

```
In [ ]: X[1, 2] = 17
        X
```

```
Out[ ]: tensor([[ 0.,  1.,  2.,  3.],
                [ 4.,  5., 17.,  7.],
                [ 8.,  9., 10., 11.]])
```

```
In [ ]: X[:2, :] = 12
        X
```

```
Out[ ]: tensor([[12., 12., 12., 12.],
                [12., 12., 12., 12.],
                [ 8.,  9., 10., 11.]])
```

2.1.3 Operations

```
In [ ]: torch.exp(x)
```

```
Out[ ]: tensor([162754.7969, 162754.7969, 162754.7969, 162754.7969, 162754.7969,
                162754.7969, 162754.7969, 162754.7969,   2980.9580,   8103.0840,
                 22026.4648,  59874.1406])
```

```
In [ ]: x = torch.tensor([1.0, 2, 4, 8])
        y = torch.tensor([2, 2, 2, 2])
        x + y, x - y, x * y, x / y, x ** y
```

```
Out[ ]: (tensor([ 3.,  4.,  6., 10.]),
         tensor([-1.,  0.,  2.,  6.]),
         tensor([ 2.,  4.,  8., 16.]),
         tensor([0.5000, 1.0000, 2.0000, 4.0000]),
         tensor([ 1.,  4., 16., 64.]))
```

```
In [ ]: X = torch.arange(12, dtype=torch.float32).reshape((3,4))
        Y = torch.tensor([[2.0, 1, 4, 3], [1, 2, 3, 4], [4, 3, 2, 1]])
        torch.cat((X, Y), dim=0), torch.cat((X, Y), dim=1)
```

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

```
In [ ]: X == Y
```

```
Out[ ]: tensor([[ True, False, False, False],
                [False, False, False, False],
                [False, False, False, False]])
```

```
In [ ]: X.sum()
```

```
Out[ ]: tensor(66.)
```

2.1.4. Broadcasting

```
In [ ]: a = torch.arange(3).reshape((3, 1))
        b = torch.arange(2).reshape((1, 2))
        a, b
```

```
Out[ ]: (tensor([[0],
                 [1],
                 [2]]),
         tensor([[0, 1]]))
```

```
In [ ]: a + b
```

```
Out[ ]: tensor([[0, 1],
                [1, 2],
                [2, 3]])
```

2.1.5. Saving Memory

```
In [ ]: before = id(Y)
        Y = Y + X
        id(Y) == before
```

```
Out[ ]: False
```

```
In [ ]: Z = torch.zeros_like(Y)
        print('id(Z):', id(Z))
```

```
Z[:] = X + Y
print('id(Z):', id(Z))
```

```
id(Z): 140394064981632
id(Z): 140394064981632
```

In [ ]:
```
before = id(X)
X += Y
id(X) == before
```

Out[ ]:
```
True
```

2.1.6. Conversion to Other Python Objects

In [ ]:
```
A = X.numpy()
B = torch.from_numpy(A)
type(A), type(B)
```

Out[ ]:
```
(numpy.ndarray, torch.Tensor)
```

In [ ]:
```
a = torch.tensor([3.5])
a, a.item(), float(a), int(a)
```

Out[ ]:
```
(tensor([3.5000]), 3.5, 3.5, 3)
```

2.2. Data Preprocessing

2.2.1. Reading The Dataset

In [ ]:
```
import os

os.makedirs(os.path.join('..', 'data'), exist_ok=True)
data_file = os.path.join('..', 'data', 'house_tiny.csv')
with open(data_file, 'w') as f:
    f.write('''NumRooms,RoofType,Price
NA,NA,127500
2,NA,106000
4,Slate,178100
NA,NA,140000''')
```

In [ ]:
```
import pandas as pd

data = pd.read_csv(data_file)
print(data)
```

```
   NumRooms RoofType   Price
0       NaN      NaN  127500
1       2.0      NaN  106000
2       4.0    Slate  178100
3       NaN      NaN  140000
```

2.2.2. Data Preparation

In [ ]:
```
inputs, targets = data.iloc[:, 0:2], data.iloc[:, 2]
inputs = pd.get_dummies(inputs, dummy_na=True)
print(inputs)
```

```
   NumRooms  RoofType_Slate  RoofType_nan
0       NaN           False          True
1       2.0           False          True
2       4.0            True         False
3       NaN           False          True
```

In [ ]:
```
inputs = inputs.fillna(inputs.mean())
print(inputs)
```

```
   NumRooms  RoofType_Slate  RoofType_nan
0       3.0           False          True
1       2.0           False          True
2       4.0            True         False
3       3.0           False          True
```

2.2.3. Conversion to the Tensor Format

In [ ]:
```
import torch

X = torch.tensor(inputs.to_numpy(dtype=float))
y = torch.tensor(targets.to_numpy(dtype=float))
X, y
```

Out[ ]:
```
(tensor([[3., 0., 1.],
         [2., 0., 1.],
         [4., 1., 0.],
         [3., 0., 1.]], dtype=torch.float64),
 tensor([127500., 106000., 178100., 140000.], dtype=torch.float64))
```

2.3. Linear Algebra

In [ ]:
```
import torch
```

2.3.1. Scalars

In [ ]:
```
x = torch.tensor(3.0)
y = torch.tensor(2.0)

x + y, x * y, x / y, x**y
```

Out[ ]:
```
(tensor(5.), tensor(6.), tensor(1.5000), tensor(9.))
```

In [ ]:
```
x = torch.arange(3)
```

```
x
```

```
Out[ ]: tensor([0, 1, 2])
```

```
In [ ]: x[2]
```

```
Out[ ]: tensor(2)
```

```
In [ ]: len(x)
```

```
Out[ ]: 3
```

```
In [ ]: x.shape
```

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

2.3.3. Matrices

```
In [ ]: A = torch.arange(6).reshape(3, 2)
        A
```

```
Out[ ]: tensor([[0, 1],
                [2, 3],
                [4, 5]])
```

```
In [ ]: A.T
```

```
Out[ ]: tensor([[0, 2, 4],
                [1, 3, 5]])
```

```
In [ ]: A = torch.tensor([[1, 2, 3], [2, 0, 4], [3, 4, 5]])
        A == A.T
```

```
Out[ ]: tensor([[True, True, True],
                [True, True, True],
                [True, True, True]])
```

2.3.4. Tensors

```
In [ ]: torch.arange(24).reshape(2, 3, 4)
```

```
Out[ ]: tensor([[[ 0,  1,  2,  3],
                 [ 4,  5,  6,  7],
                 [ 8,  9, 10, 11]],

                [[12, 13, 14, 15],
                 [16, 17, 18, 19],
                 [20, 21, 22, 23]]])
```

```
In [ ]: A = torch.arange(6, dtype=torch.float32).reshape(2, 3)
        B = A.clone()  # Assign a copy of A to B by allocating new memory
```

```
A, A + B
```

```
Out[ ]: (tensor([[0., 1., 2.],
                 [3., 4., 5.]]),
         tensor([[ 0.,  2.,  4.],
                 [ 6.,  8., 10.]]))
```

```
In [ ]: A * B
```

```
Out[ ]: tensor([[ 0.,  1.,  4.],
                [ 9., 16., 25.]])
```

```
In [ ]: a = 2
        X = torch.arange(24).reshape(2, 3, 4)
        a + X, (a * X).shape
```

```
Out[ ]: (tensor([[[ 2,  3,  4,  5],
                  [ 6,  7,  8,  9],
                  [10, 11, 12, 13]],

                 [[14, 15, 16, 17],
                  [18, 19, 20, 21],
                  [22, 23, 24, 25]]]),
         torch.Size([2, 3, 4]))
```

2.3.6. Reduction

```
In [ ]: x = torch.arange(3, dtype=torch.float32)
        x, x.sum()
```

```
Out[ ]: (tensor([0., 1., 2.]), tensor(3.))
```

```
In [ ]: A.shape, A.sum()
```

```
Out[ ]: (torch.Size([2, 3]), tensor(15.))
```

```
In [ ]: A.shape, A.sum(axis=0).shape
```

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

```
In [ ]: A.shape, A.sum(axis=1).shape
```

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

```
In [ ]: A.sum(axis=[0, 1]) == A.sum()  # Same as A.sum()
```

```
Out[ ]: tensor(True)
```

```
In [ ]: A.mean(), A.sum() / A.numel()
```

```
Out[ ]: (tensor(2.5000), tensor(2.5000))
```

```
In [ ]: A.mean(axis=0), A.sum(axis=0) / A.shape[0]

Out[ ]: (tensor([1.5000, 2.5000, 3.5000]), tensor([1.5000, 2.5000, 3.5000]))
```

2.3.7. Non-Reduction Sum

```
In [ ]: sum_A = A.sum(axis=1, keepdims=True)
        sum_A, sum_A.shape

Out[ ]: (tensor([[ 3.],
                 [12.]]),
         torch.Size([2, 1]))

In [ ]: A / sum_A

Out[ ]: tensor([[0.0000, 0.3333, 0.6667],
                [0.2500, 0.3333, 0.4167]])
```

2.3.8. Dot Products

```
In [ ]: y = torch.ones(3, dtype = torch.float32)
        x, y, torch.dot(x, y)

Out[ ]: (tensor([0., 1., 2.]), tensor([1., 1., 1.]), tensor(3.))
```

2.3.9. Matrix-Vector Products

```
In [ ]: A.shape, x.shape, torch.mv(A, x), A@x

Out[ ]: (torch.Size([2, 3]), torch.Size([3]), tensor([ 5., 14.]), tensor([ 5., 1
        4.]))
```

2.3.10. Matrix-Matrix Multiplication

```
In [ ]: B = torch.ones(3, 4)
        torch.mm(A, B), A@B

Out[ ]: (tensor([[ 3.,  3.,  3.,  3.],
                 [12., 12., 12., 12.]]),
         tensor([[ 3.,  3.,  3.,  3.],
                 [12., 12., 12., 12.]]))
```

2.3.11. Norms

```
In [ ]: u = torch.tensor([3.0, -4.0])
        torch.norm(u)
```

```
Out[ ]: tensor(5.)
```

```
In [ ]: torch.norm(torch.ones((4, 9)))
```

```
Out[ ]: tensor(6.)
```

2.5. Automatic Differentiation

```
In [ ]: import torch
```

2.5.1. A Simple Function

```
In [ ]: x = torch.arange(4.0)
        x
```

```
Out[ ]: tensor([0., 1., 2., 3.])
```

```
In [ ]: x.requires_grad_(True)
        x.grad
```

```
In [ ]: y = 2 * torch.dot(x, x)
        y
```

```
Out[ ]: tensor(28., grad_fn=<MulBackward0>)
```

```
In [ ]: y.backward()
        x.grad
```

```
Out[ ]: tensor([ 0.,  4.,  8., 12.])
```

```
In [ ]: x.grad.zero_()  # Reset the gradient
        y = x.sum()
        y.backward()
        x.grad
```

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

2.5.2. Backward for Non-Scalar Variables

```
In [ ]: x.grad.zero_()
        y = x * x
        y.backward(gradient=torch.ones(len(y)))  # Faster: y.sum().backward()
        x.grad
```

```
Out[ ]: tensor([0., 2., 4., 6.])
```

2.5.3. Dataching Computation

In [ ]:
```
x.grad.zero_()
y = x * x
u = y.detach()
z = u * x

z.sum().backward()
x.grad == u
```

Out[ ]:
```
tensor([True, True, True, True])
```

In [ ]:
```
x.grad.zero_()
y.sum().backward()
x.grad == 2 * x
```

Out[ ]:
```
tensor([True, True, True, True])
```

2.5.4. Gardients and Python Control Flow

In [ ]:
```
def f(a):
    b = a * 2
    while b.norm() < 1000:
        b = b * 2
    if b.sum() > 0:
        c = b
    else:
        c = 100 * b
    return c
```

In [ ]:
```
a = torch.randn(size=(), requires_grad=True)
d = f(a)
d.backward()
```

In [ ]:
```
a.grad == d / a
```

Out[ ]:
```
tensor(True)
```

3.1. Linear Regression

In [ ]:
```
%matplotlib inline
import math
import time
import numpy as np
import torch
from d2l import torch as d2l
```

3.1.2. Vectorization for Speed

In [ ]:
```
n = 10000
a = torch.ones(n)
b = torch.ones(n)
```

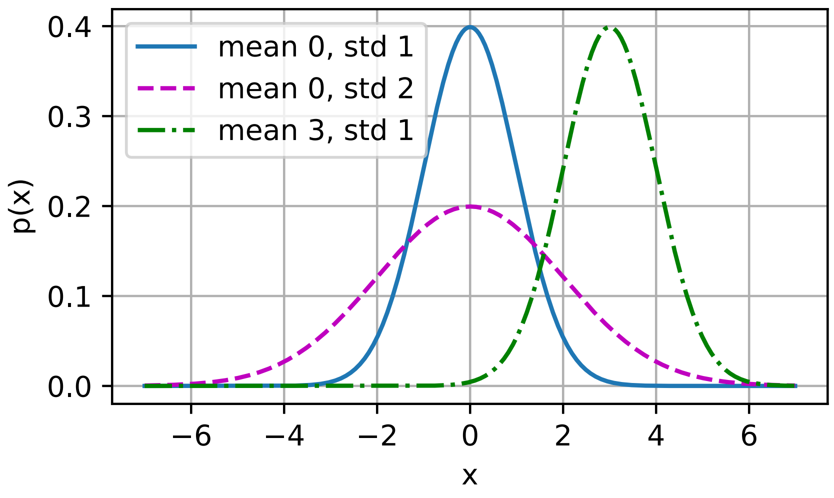
In [ ]:
```
c = torch.zeros(n)
t = time.time()
for i in range(n):
    c[i] = a[i] + b[i]
f'{time.time() - t:.5f} sec'
```

Out[ ]:
```
'0.18140 sec'
```

3.1.3. The Normal Distribution and Squared Loss

In [ ]:
```
def normal(x, mu, sigma):
    p = 1 / math.sqrt(2 * math.pi * sigma**2)
    return p * np.exp(-0.5 * (x - mu)**2 / sigma**2)
```

In [ ]:
```
x = np.arange(-7, 7, 0.01)
params = [(0, 1), (0, 2), (3, 1)]
d2l.plot(x, [normal(x, mu, sigma) for mu, sigma in params], xlabel='x',
         ylabel='p(x)', figsize=(4.5, 2.5),
         legend=[f'mean {mu}, std {sigma}' for mu, sigma in params])
```



3.2. Object-Oriented Design for Implementation

In [ ]:
```
import time
import numpy as np
import torch
from torch import nn
from d2l import torch as d2l
```

2.3.1. Utilities

In [ ]:
```python
def add_to_class(Class):
    """Register functions as methods in created class."""
    def wrapper(obj):
        setattr(Class, obj.__name__, obj)
    return wrapper
```

In [ ]:
```python
class A:
    def __init__(self):
        self.b = 1

a = A()
```

In [ ]:
```python
@add_to_class(A)
def do(self):
    print('Class attribute "b" is', self.b)

a.do()
```

```
Class attribute "b" is 1
```

In [ ]:
```python
class HyperParameters:
  def save_hyperparameters(self, ignore=[]):
        raise NotImplemented
```

In [ ]:
```python
class B(d2l.HyperParameters):
    def __init__(self, a, b, c):
        self.save_hyperparameters(ignore=['c'])
        print('self.a =', self.a, 'self.b =', self.b)
        print('There is no self.c =', not hasattr(self, 'c'))

b = B(a=1, b=2, c=3)
```

```
self.a = 1 self.b = 2
There is no self.c = True
```

3.2.2. Models

In [ ]:
```python
class Module(nn.Module, d2l.HyperParameters):
    def __init__(self, plot_train_per_epoch=2, plot_valid_per_epoch=1):
        super().__init__()
        self.save_hyperparameters()
        self.board = ProgressBoard()

    def loss(self, y_hat, y):
        raise NotImplementedError

    def forward(self, X):
        assert hasattr(self, 'net'), 'Neural network is defined'
        return self.net(X)

    def plot(self, key, value, train):
        assert hasattr(self, 'trainer'), 'Trainer is not inited'
        self.board.xlabel = 'epoch'
        if train:
            x = self.trainer.train_batch_idx / \
                self.trainer.num_train_batches
            n = self.trainer.num_train_batches / \
                self.plot_train_per_epoch
        else:
            x = self.trainer.epoch + 1
            n = self.trainer.num_val_batches / \
                self.plot_valid_per_epoch
        self.board.draw(x, value.to(d2l.cpu()).detach().numpy(),
                        ('train_' if train else 'val_') + key,
                        every_n=int(n))

    def training_step(self, batch):
        l = self.loss(self(*batch[:-1]), batch[-1])
        self.plot('loss', l, train=True)
        return l

    def validation_step(self, batch):
        l = self.loss(self(*batch[:-1]), batch[-1])
        self.plot('loss', l, train=False)

    def configure_optimizers(self):
        raise NotImplementedError
```

3.2.3. Data

In [ ]:
```
class DataModule(d2l.HyperParameters):
    def __init__(self, root='../data', num_workers=4):
        self.save_hyperparameters()

    def get_dataloader(self, train):
        raise NotImplementedError

    def train_dataloader(self):
        return self.get_dataloader(train=True)

    def val_dataloader(self):
        return self.get_dataloader(train=False)
```

3.2.4. Training

In [ ]:
```
class Trainer(d2l.HyperParameters):
    def __init__(self, max_epochs, num_gpus=0, gradient_clip_val=0):
        self.save_hyperparameters()
        assert num_gpus == 0, 'No GPU support yet'

    def prepare_data(self, data):
        self.train_dataloader = data.train_dataloader()
        self.val_dataloader = data.val_dataloader()
        self.num_train_batches = len(self.train_dataloader)
        self.num_val_batches = (len(self.val_dataloader)
                                if self.val_dataloader is not None else 0

    def prepare_model(self, model):
        model.trainer = self
        model.board.xlim = [0, self.max_epochs]
        self.model = model

    def fit(self, model, data):
        self.prepare_data(data)
        self.prepare_model(model)
        self.optim = model.configure_optimizers()
        self.epoch = 0
        self.train_batch_idx = 0
        self.val_batch_idx = 0
        for self.epoch in range(self.max_epochs):
            self.fit_epoch()

    def fit_epoch(self):
        raise NotImplementedError
```

3.4. Linear Regression Implementation from Scratch

In [ ]:
```
%matplotlib inline
import torch
from d2l import torch as d2l
```

3.4.1. Defining the Model

In [ ]:
```
class LinearRegressionScratch(d2l.Module):
    def __init__(self, num_inputs, lr, sigma=0.01):
        super().__init__()
        self.save_hyperparameters()
        self.w = torch.normal(0, sigma, (num_inputs, 1), requires_grad=Tr
        self.b = torch.zeros(1, requires_grad=True)
```

In [ ]:
```
@d2l.add_to_class(LinearRegressionScratch)
def forward(self, X):
    return torch.matmul(X, self.w) + self.b
```

3.4.2. Defining the Loss Function

In [ ]:
```
@d2l.add_to_class(LinearRegressionScratch)
def loss(self, y_hat, y):
    l = (y_hat - y) ** 2 / 2
    return l.mean()
```

3.4.3. Defining the Optimization Algorithm

In [ ]:
```
class SGD(d2l.HyperParameters):
    def __init__(self, params, lr):
        self.save_hyperparameters()

    def step(self):
        for param in self.params:
            param -= self.lr * param.grad

    def zero_grad(self):
        for param in self.params:
            if param.grad is not None:
                param.grad.zero_()
```

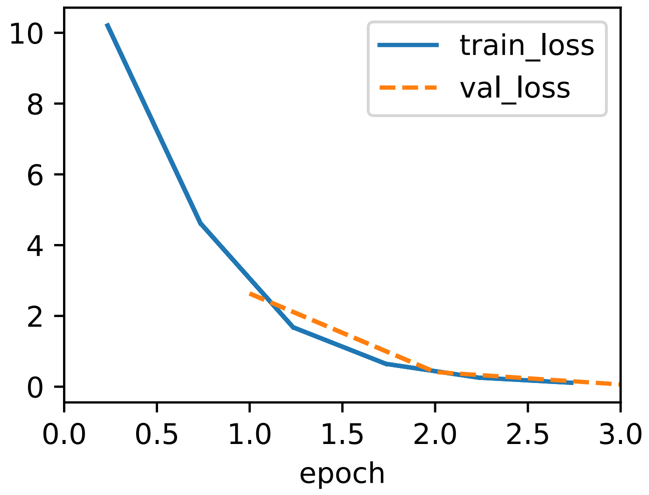
In [ ]:
```
@d2l.add_to_class(LinearRegressionScratch)
def configure_optimizers(self):
    return SGD([self.w, self.b], self.lr)
```

3.4.4. Training

In [ ]:
```
@d2l.add_to_class(d2l.Trainer)
def prepare_batch(self, batch):
    return batch

@d2l.add_to_class(d2l.Trainer)
def fit_epoch(self):
    self.model.train()
    for batch in self.train_dataloader:
        loss = self.model.training_step(self.prepare_batch(batch))
        self.optim.zero_grad()
        with torch.no_grad():
            loss.backward()
            if self.gradient_clip_val > 0:
                self.clip_gradients(self.gradient_clip_val, self.model)
            self.optim.step()
        self.train_batch_idx += 1
    if self.val_dataloader is None:
        return
    self.model.eval()
    for batch in self.val_dataloader:
        with torch.no_grad():
            self.model.validation_step(self.prepare_batch(batch))
        self.val_batch_idx += 1
```

In [ ]:
```
model = LinearRegressionScratch(2, lr=0.03)
data = d2l.SyntheticRegressionData(w=torch.tensor([2, -3.4]), b=4.2)
trainer = d2l.Trainer(max_epochs=3)
trainer.fit(model, data)
```



In [ ]:
```
with torch.no_grad():
    print(f'error in estimating w: {data.w - model.w.reshape(data.w.shape
    print(f'error in estimating b: {data.b - model.b}')
```

```
error in estimating w: tensor([ 0.1033, -0.2273])
error in estimating b: tensor([0.2499])
```

3.4. Linear Regression Implementation from Scratch

3.4.1. Defining the Model

In [ ]:
```
%matplotlib inline
import torch
from d2l import torch as d2l
```

In [ ]:
```
class LinearRegressionScratch(d2l.Module):
    def __init__(self, num_inputs, lr, sigma=0.01):
        super().__init__()
        self.save_hyperparameters()
        self.w = torch.normal(0, sigma, (num_inputs, 1), requires_grad=Tr
        self.b = torch.zeros(1, requires_grad=True)
```

In [ ]:
```
@d2l.add_to_class(LinearRegressionScratch)
def forward(self, X):
    return torch.matmul(X, self.w) + self.b
```

3.4.2. Defining the Loss Function

In [ ]:
```
@d2l.add_to_class(LinearRegressionScratch)
def loss(self, y_hat, y):
    l = (y_hat - y) ** 2 / 2
    return l.mean()
```

3.4.3. Defining the Optimization Algorithm

In [ ]:
```
class SGD(d2l.HyperParameters):
    def __init__(self, params, lr):
        self.save_hyperparameters()

    def step(self):
        for param in self.params:
            param -= self.lr * param.grad

    def zero_grad(self):
        for param in self.params:
            if param.grad is not None:
                param.grad.zero_()
```

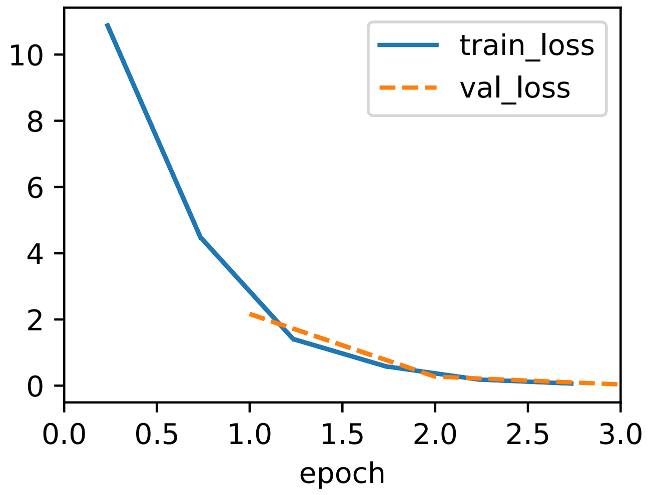
In [ ]:
```
@d2l.add_to_class(LinearRegressionScratch)
def configure_optimizers(self):
    return SGD([self.w, self.b], self.lr)
```

3.4.4. Training

In [ ]:
```
@d2l.add_to_class(d2l.Trainer)
def prepare_batch(self, batch):
    return batch

@d2l.add_to_class(d2l.Trainer)
def fit_epoch(self):
    self.model.train()
    for batch in self.train_dataloader:
        loss = self.model.training_step(self.prepare_batch(batch))
        self.optim.zero_grad()
        with torch.no_grad():
            loss.backward()
            if self.gradient_clip_val > 0:
                self.clip_gradients(self.gradient_clip_val, self.model)
            self.optim.step()
        self.train_batch_idx += 1
    if self.val_dataloader is None:
        return
    self.model.eval()
    for batch in self.val_dataloader:
        with torch.no_grad():
            self.model.validation_step(self.prepare_batch(batch))
        self.val_batch_idx += 1
```

In [ ]:
```
model = LinearRegressionScratch(2, lr=0.03)
data = d2l.SyntheticRegressionData(w=torch.tensor([2, -3.4]), b=4.2)
trainer = d2l.Trainer(max_epochs=3)
trainer.fit(model, data)
```



In [ ]:
```
with torch.no_grad():
    print(f'error in estimating w: {data.w - model.w.reshape(data.w.shape
    print(f'error in estimating b: {data.b - model.b}')
```

```
error in estimating w: tensor([ 0.0767, -0.1514])
error in estimating b: tensor([0.2047])
```

4.1. Softmax Regression

4.2. The Image Classification Dataset

In [ ]:
```
%matplotlib inline
import time
import torch
import torchvision
from torchvision import transforms
from d2l import torch as d2l

d2l.use_svg_display()
```

4.2.1. Loading the Dataset

In [ ]:
```
class FashionMNIST(d2l.DataModule):
    def __init__(self, batch_size=64, resize=(28, 28)):
        super().__init__()
        self.save_hyperparameters()
        trans = transforms.Compose([transforms.Resize(resize),
                                    transforms.ToTensor()])
        self.train = torchvision.datasets.FashionMNIST(
            root=self.root, train=True, transform=trans, download=True)
        self.val = torchvision.datasets.FashionMNIST(
            root=self.root, train=False, transform=trans, download=True)
```

In [ ]:
```
data = FashionMNIST(resize=(32, 32))
```

```
len(data.train), len(data.val)
```

```
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/tra
in-images-idx3-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/tra
in-images-idx3-ubyte.gz to ../data/FashionMNIST/raw/train-images-idx3-ubyt
e.gz
100%|██████████| 26421880/26421880 [00:02<00:00, 12340574.74it/s]
Extracting ../data/FashionMNIST/raw/train-images-idx3-ubyte.gz to ../data/
FashionMNIST/raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/tra
in-labels-idx1-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/tra
in-labels-idx1-ubyte.gz to ../data/FashionMNIST/raw/train-labels-idx1-ubyt
e.gz
100%|██████████| 29515/29515 [00:00<00:00, 199828.71it/s]
Extracting ../data/FashionMNIST/raw/train-labels-idx1-ubyte.gz to ../data/
FashionMNIST/raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10
k-images-idx3-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10
k-images-idx3-ubyte.gz to ../data/FashionMNIST/raw/t10k-images-idx3-ubyte.
gz
100%|██████████| 4422102/4422102 [00:03<00:00, 1466258.57it/s]
Extracting ../data/FashionMNIST/raw/t10k-images-idx3-ubyte.gz to ../data/F
ashionMNIST/raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10
k-labels-idx1-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10
k-labels-idx1-ubyte.gz to ../data/FashionMNIST/raw/t10k-labels-idx1-ubyte.
gz
100%|██████████| 5148/5148 [00:00<00:00, 4764403.57it/s]
Extracting ../data/FashionMNIST/raw/t10k-labels-idx1-ubyte.gz to ../data/F
ashionMNIST/raw
```

```
Out[ ]:  (60000, 10000)
```

```
In [ ]:  data.train[0][0].shape
```

```
Out[ ]:  torch.Size([1, 32, 32])
```

```
In [ ]:  @d2l.add_to_class(FashionMNIST)
         def text_labels(self, indices):
             labels = ['t-shirt', 'trouser', 'pullover', 'dress', 'coat',
                       'sandal', 'shirt', 'sneaker', 'bag', 'ankle boot']
             return [labels[int(i)] for i in indices]
```

4.2.2. Reading a Minibatch

```
In [ ]: @d2l.add_to_class(FashionMNIST)
        def get_dataloader(self, train):
            data = self.train if train else self.val
            return torch.utils.data.DataLoader(data, self.batch_size, shuffle=tra
                                               num_workers=self.num_workers)
```

```
In [ ]: X, y = next(iter(data.train_dataloader()))
        print(X.shape, X.dtype, y.shape, y.dtype)
```

```
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:55
7: UserWarning: This DataLoader will create 4 worker processes in total. O
ur suggested max number of worker in current system is 2, which is smaller
than what this DataLoader is going to create. Please be aware that excessi
ve worker creation might get DataLoader running slow or even freeze, lower
the worker number to avoid potential slowness/freeze if necessary.
  warnings.warn(_create_warning_msg(
torch.Size([64, 1, 32, 32]) torch.float32 torch.Size([64]) torch.int64
```

```
In [ ]: tic = time.time()
        for X, y in data.train_dataloader():
            continue
        f'{time.time() - tic:.2f} sec'

Out[ ]: '16.20 sec'
```

4.2.3. Visualization

```
In [ ]: def show_images(imgs, num_rows, num_cols, titles=None, scale=1.5):
            raise NotImplementedError
```

```
In [ ]: @d2l.add_to_class(FashionMNIST)
        def visualize(self, batch, nrows=1, ncols=8, labels=[]):
            X, y = batch
            if not labels:
                labels = self.text_labels(y)
            d2l.show_images(X.squeeze(1), nrows, ncols, titles=labels)
        batch = next(iter(data.val_dataloader()))
        data.visualize(batch)
```

4.3. The Base Classification Model

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

4.3.1. The Classifier Class

In [ ]:
```
class Classifier(d2l.Module):
    def validation_step(self, batch):
        Y_hat = self(*batch[:-1])
        self.plot('loss', self.loss(Y_hat, batch[-1]), train=False)
        self.plot('acc', self.accuracy(Y_hat, batch[-1]), train=False)
```

In [ ]:
```
@d2l.add_to_class(d2l.Module)
def configure_optimizers(self):
    return torch.optim.SGD(self.parameters(), lr=self.lr)
```

4.3.2. Accuracy

In [ ]:
```
@d2l.add_to_class(Classifier)
def accuracy(self, Y_hat, Y, averaged=True):
    Y_hat = Y_hat.reshape((-1, Y_hat.shape[-1]))
    preds = Y_hat.argmax(axis=1).type(Y.dtype)
    compare = (preds == Y.reshape(-1)).type(torch.float32)
    return compare.mean() if averaged else compare
```

4.4. Softmax Regression Implementation from Scratch

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

4.4.1. The Softmax

In [ ]:
```
X = torch.tensor([[1.0, 2.0, 3.0], [4.0, 5.0, 6.0]])
X.sum(0, keepdims=True), X.sum(1, keepdims=True)
```

Out[ ]:
```
(tensor([[5., 7., 9.]]),
 tensor([[ 6.],
         [15.]]))
```

In [ ]:
```
def softmax(X):
    X_exp = torch.exp(X)
    partition = X_exp.sum(1, keepdims=True)
    return X_exp / partition
```

In [ ]:
```
X = torch.rand((2, 5))
X_prob = softmax(X)
X_prob, X_prob.sum(1)
```

Out[ ]:
```
(tensor([[0.1520, 0.1707, 0.1840, 0.2317, 0.2617],
         [0.2110, 0.2008, 0.2523, 0.1399, 0.1960]]),
 tensor([1.0000, 1.0000]))
```

4.4.2. The Model

In [ ]:
```
class SoftmaxRegressionScratch(d2l.Classifier):
    def __init__(self, num_inputs, num_outputs, lr, sigma=0.01):
        super().__init__()
        self.save_hyperparameters()
        self.W = torch.normal(0, sigma, size=(num_inputs, num_outputs),
                              requires_grad=True)
        self.b = torch.zeros(num_outputs, requires_grad=True)

    def parameters(self):
        return [self.W, self.b]
```

In [ ]:
```
@d2l.add_to_class(SoftmaxRegressionScratch)
def forward(self, X):
    X = X.reshape((-1, self.W.shape[0]))
    return softmax(torch.matmul(X, self.W) + self.b)
```

4.4.3. The Cross-Entropy Loss

In [ ]:
```
y = torch.tensor([0, 2])
y_hat = torch.tensor([[0.1, 0.3, 0.6], [0.3, 0.2, 0.5]])
y_hat[[0, 1], y]
```

Out[ ]:
```
tensor([0.1000, 0.5000])
```

In [ ]:
```
def cross_entropy(y_hat, y):
    return -torch.log(y_hat[list(range(len(y_hat))), y]).mean()

cross_entropy(y_hat, y)
```

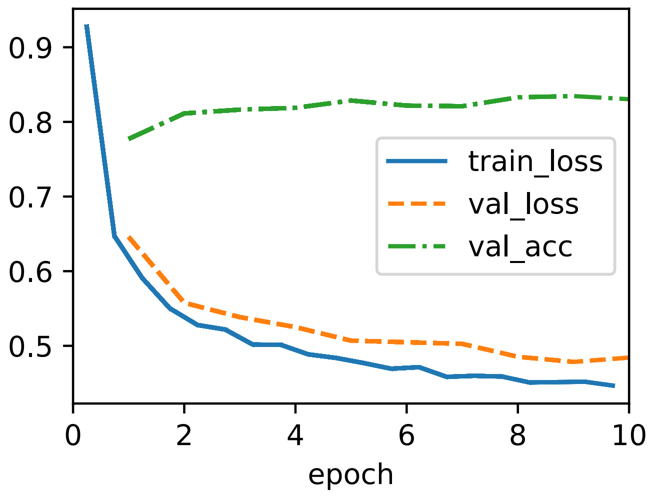
Out[ ]:
```
tensor(1.4979)
```

In [ ]:
```
@d2l.add_to_class(SoftmaxRegressionScratch)
def loss(self, y_hat, y):
    return cross_entropy(y_hat, y)
```

4.4.4. Training

In [ ]:
```
data = d2l.FashionMNIST(batch_size=256)
model = SoftmaxRegressionScratch(num_inputs=784, num_outputs=10, lr=0.1)
```

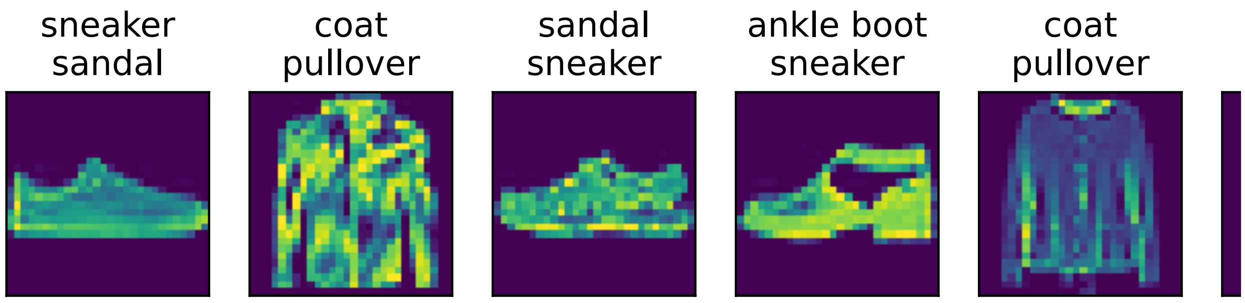
```
trainer = d2l.Trainer(max_epochs=10)
trainer.fit(model, data)
```



In [ ]:
```
X, y = next(iter(data.val_dataloader()))
preds = model(X).argmax(axis=1)
preds.shape
```

Out[ ]:
```
torch.Size([256])
```

In [ ]:
```
wrong = preds.type(y.dtype) != y
X, y, preds = X[wrong], y[wrong], preds[wrong]
labels = [a+'\n'+b for a, b in zip(
    data.text_labels(y), data.text_labels(preds))]
data.visualize([X, y], labels=labels)
```



5.1. Multilayer Perceptrons

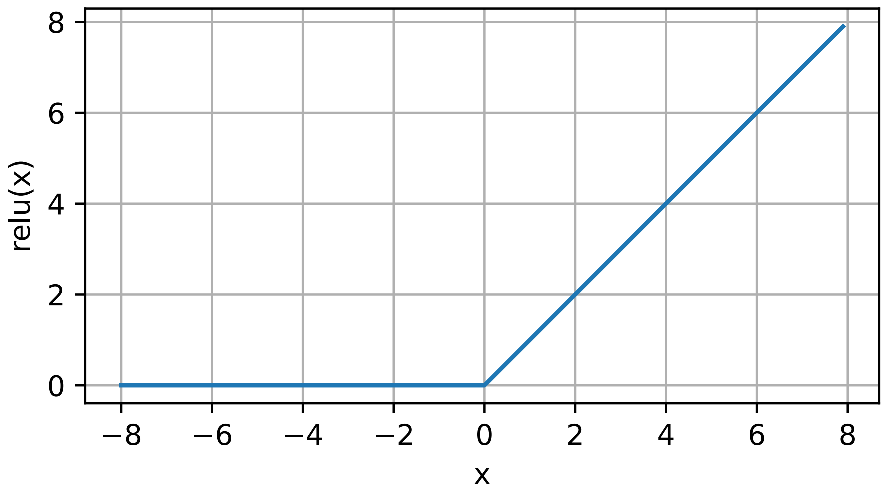
In [ ]:
```
%matplotlib inline
```

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

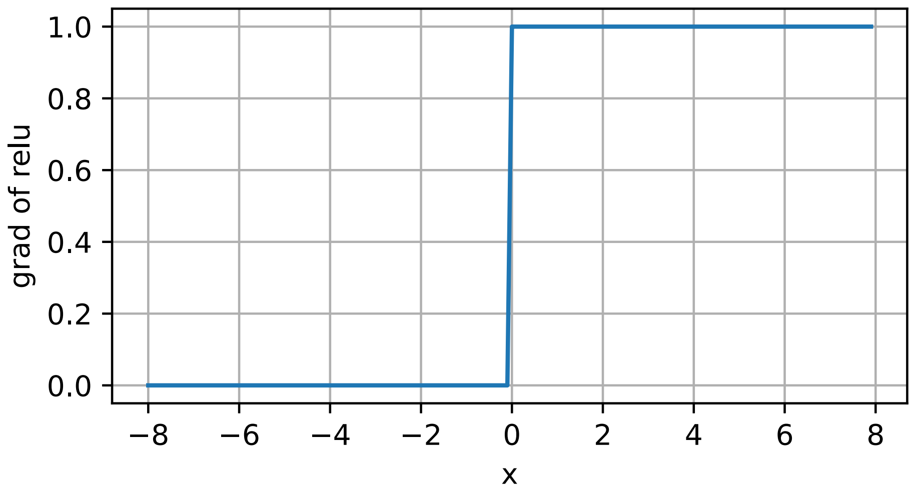
$$\mathrm{ReLU}(x) = \max(x, 0)$$

In [ ]:
```
x=torch.arange(-8.0,8.0,0.1,requires_grad=True)
```

```
y=torch.relu(x)
d2l.plot(x.detach(),y.detach(),'x','relu(x)',figsize=(5,2.5))
```

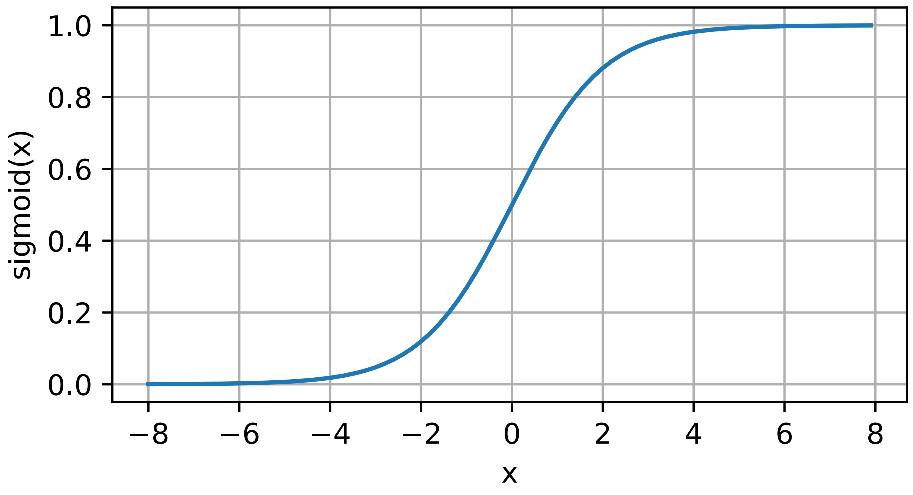


In [ ]:
```
y.backward(torch.ones_like(x),retain_graph=True)
d2l.plot(x.detach(),x.grad,'x','grad of relu',figsize=(5,2.5))
```

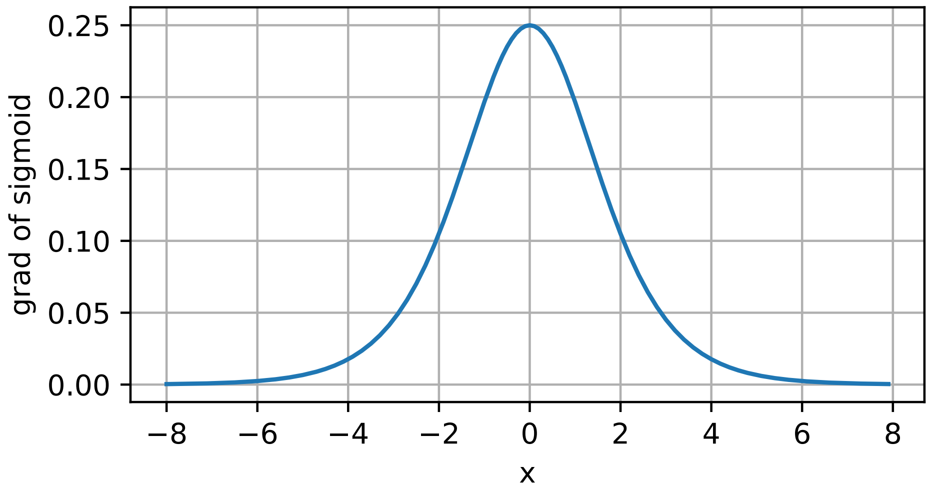


Continued

In [ ]:
```
y=torch.sigmoid(x)
d2l.plot(x.detach(),y.detach(),'x','sigmoid(x)',figsize=(5,2.5))
```



In [ ]:
```
x.grad.data.zero_()
y.backward(torch.ones_like(x),retain_graph=True)
d2l.plot(x.detach(),x.grad,'x','grad of sigmoid',figsize=(5,2.5))
```



- Gradient Vanishing Problem ~ Backpropagation / Activation functions

5.2. Implementation of Multilayer Perceptrons

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

5.2.1. Implementation from Scratch

5.2.1.1. Initializing Model Parameters

In [ ]:
```
class MLPScratch(d2l.Classifier):
    def __init__(self, num_inputs, num_outputs, num_hiddens, lr, sigma=0.
        super().__init__()
        self.save_hyperparameters()
        self.W1 = nn.Parameter(torch.randn(num_inputs, num_hiddens) * sig
        self.b1 = nn.Parameter(torch.zeros(num_hiddens))
        self.W2 = nn.Parameter(torch.randn(num_hiddens, num_outputs) * si
        self.b2 = nn.Parameter(torch.zeros(num_outputs))
```

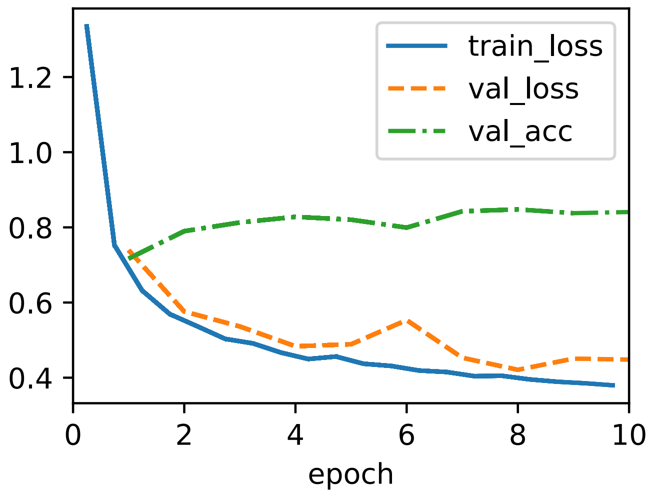
5.2.1.2. Model

In [ ]:
```
def relu(X):
    a = torch.zeros_like(X)
    return torch.max(X, a)
```

In [ ]:
```
@d2l.add_to_class(MLPScratch)
def forward(self, X):
    X = X.reshape((-1, self.num_inputs))
    H = relu(torch.matmul(X, self.W1) + self.b1)
    return torch.matmul(H, self.W2) + self.b2
```

5.2.1.3. Training

In [ ]:
```
model = MLPScratch(num_inputs=784, num_outputs=10, num_hiddens=256, lr=0.
data = d2l.FashionMNIST(batch_size=256)
trainer = d2l.Trainer(max_epochs=10)
trainer.fit(model, data)
```



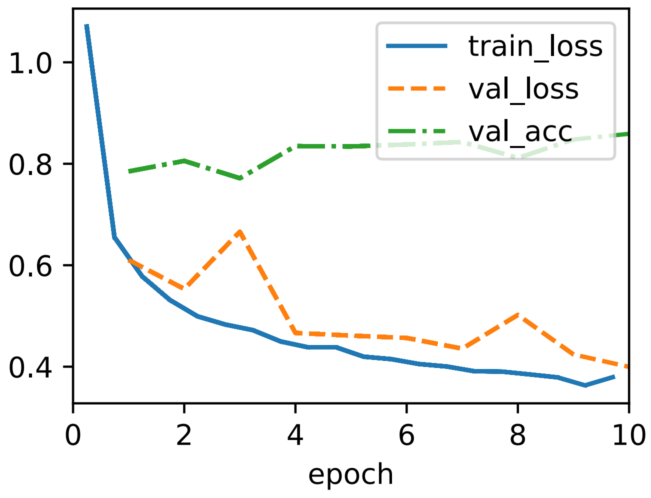
5.2.2. Concise Implementation

5.2.2.1. Model

In [ ]:
```
class MLP(d2l.Classifier):
    def __init__(self, num_outputs, num_hiddens, lr):
        super().__init__()
        self.save_hyperparameters()
        self.net = nn.Sequential(nn.Flatten(), nn.LazyLinear(num_hiddens)
                                 nn.ReLU(), nn.LazyLinear(num_outputs))
```

5.2.2.2. Training

```
In [ ]: model = MLP(num_outputs=10, num_hiddens=256, lr=0.1)
        trainer.fit(model, data)
```



5.3. Foward Propagation, Backward Propagation, and Computational Graphs

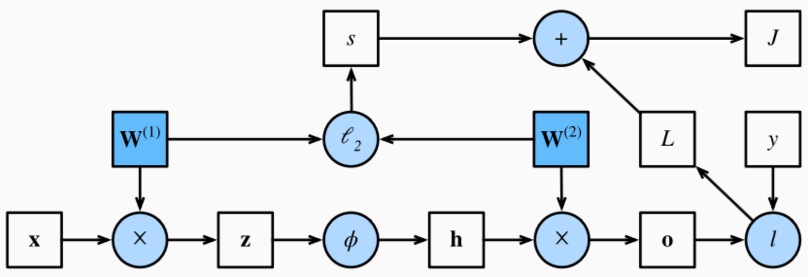


Fig. 5.3.1 Computational graph of forward propagation.

Discussions & Exercises

2.1.8 Exercise

- X<Y : tensor([[False, False, False, False], [False, False, False, False], [False, False, False, False]])
- X>Y : tensor([[False, True, True, True], [True, True, True, True], [True, True, True, True]])

2.2.4. Discussion

- Real-world datasets are often plagued by outliers, faulty measurements from sensors, and recording errors, which must be addressed before feeding the data into any model. Data visualization tools such as seaborn, Bokeh, or matplotlib can help you to manually inspect the data and develop intuitions about the type of problems you may need to address.

2.2.5. Exercises

- 4. Large number of categories can be dealt with one hot encoding. If the category labels are all unique, that model cannot predict and this cause overfitting problem. Thus, we can use clustering or feature extraction

2.3.12. Exercises

- 8. axis 0 : (3,4) axis 1 : (2,4) axis 2 : (2,3)

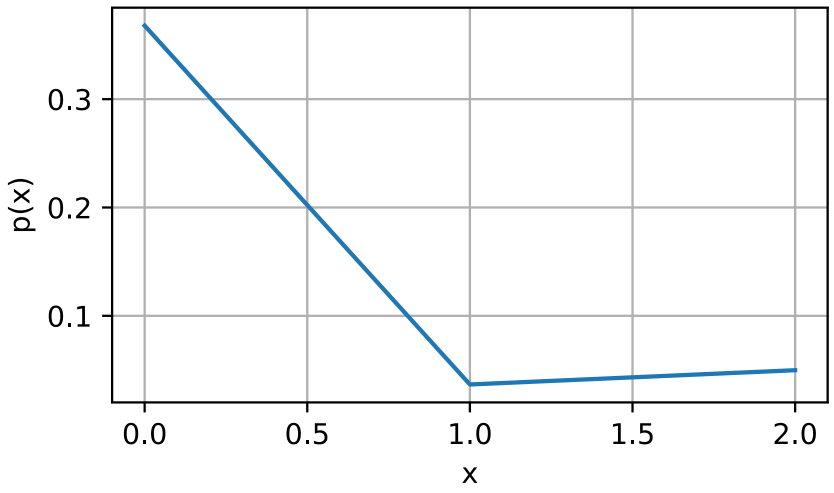
2.5.5. Discussion

- (i) attach gradients to those variables with respect to which we desire derivatives; (ii) record the computation of the target value; (iii) execute the backpropagation function; and (iv) access the resulting gradient.

3.1.6. My exercise (exponential distribution)

In [ ]:
```
def exp(x, lam):
    return lam * np.exp(-1*lam*x)
```

In [ ]:
```
params = [(1, 1), (2, 2), (3, 1)]
d2l.plot(x, [exp(x, lam) for x, lam in params], xlabel='x',
         ylabel='p(x)', figsize=(4.5, 2.5))
```



3.2.5. Discussion

4.2.5. My Exercises(reducing batch size)

- when batch size = 64, reading time = 15.19sec
- when batch size = 32, reading time = 17.47sec
- when batch size=16, reading time = 21.97sec
- when batch size=1, it took more than 2 minute

Thus, if batch size become smaller, reading time increases

To highlight the object-oriented design for our future deep learning implementation, the above classes simply show how their objects store data and interact with each other. We will keep enriching implementations of these classes, such as via @add_to_class, in the rest of the book.

4.4.7. Discussion question

Is it always a good idea to return the most likely label? For example, would you do this for medical diagnosis? How would you try to address this?

- Since in the most likely label way, there is a probability that the model gives wrong answer. Therefore, if we need very accurate model like medical diagnosis, this way might not useful.
- to address this, we can set threshold value to accuracy. For example, we can set 0.99 as a threshold value so that we can get predictions with 0.99 or higher accuracy

5.1.1.1 Memo

1.Universal Approximators

- a single-hidden-layer network, given enough nodes (possibly absurdly many), and the right set of weights, we can model any function. -kernel methods are way more effective, since they are capable of solving the problem exactly even in infinite dimensional spaces (Kimeldorf and Wahba, 1971, Schölkopf et al., 2001). "Universal Approximation Theorem" -we can approximate many functions much more compactly by using deeper (rather than wider) networks (Simonyan and Zisserman, 2014).

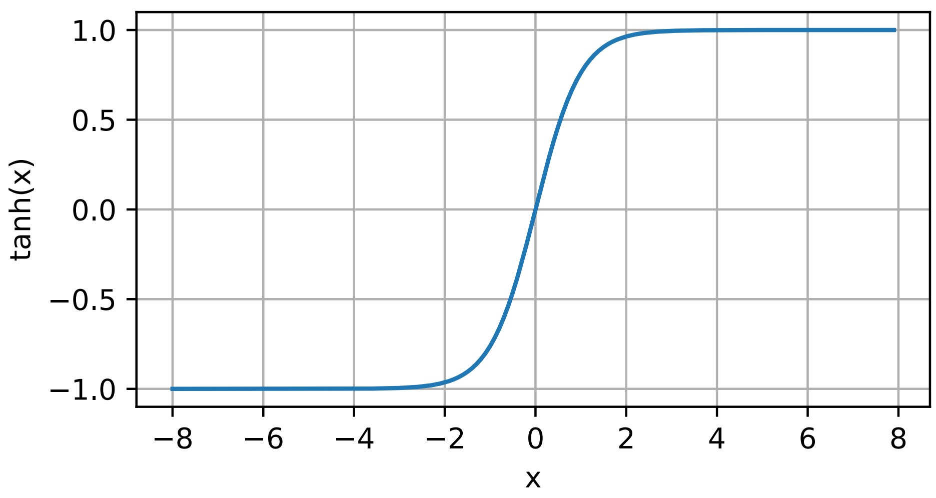
5.1.2.1 (Exercise) My own exercise/experiment

$$\mathrm{pReLU}(x) = \max(0, x) + \alpha \min(0, x)$$

5.1.2.3 Tanh Function (Discussion)

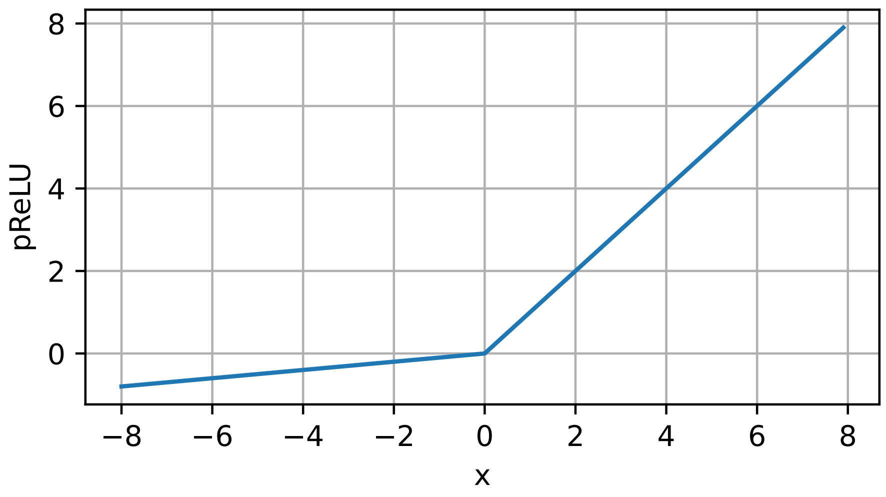
- $\tanh(x) = \frac{1-\exp(-2x)}{1+\exp(-2x)} = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$

```
In [ ]: y=torch.tanh(x)
        d2l.plot(x.detach(),y.detach(),'x','tanh(x)',figsize=(5,2.5))
```

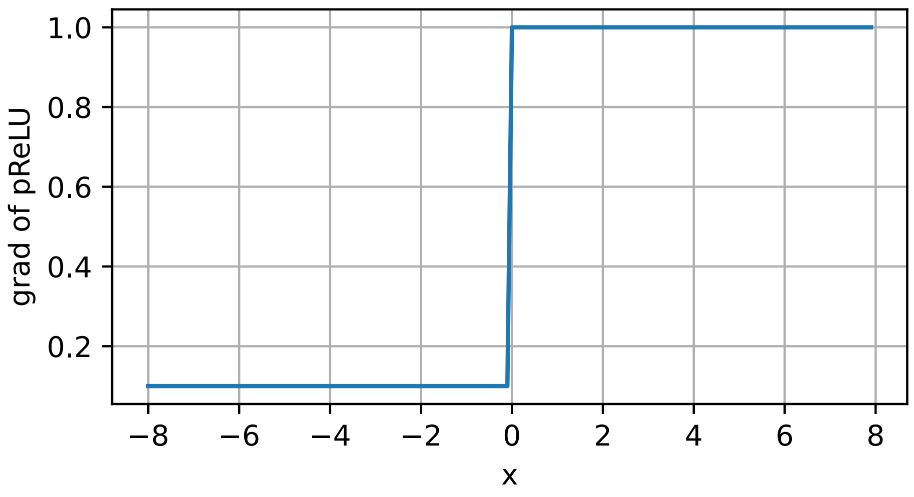


```
In [ ]: pReLU=lambda x,a:torch.max(torch.tensor(0),x)+a*torch.min(torch.tensor(0)
```

```
In [ ]: y=pReLU(x=x,a=0.1)
        d2l.plot(x.detach(),y.detach(),'x','pReLU',figsize=(5,2.5))
```



```
In [ ]: x.grad.data.zero_()
        y.backward(torch.ones_like(x),retain_graph=True)
        d2l.plot(x.detach(),x.grad,'x','grad of pReLU',figsize=(5,2.5))
```



5.2.3. My exercise(reducing hidden layer)

- when hidden layer = 256, val acc = 0.8xxx
- when hidden layer = 64, val acc = 0.75xxx

Thus when hidden layer decreases, val acc also decreases.

Also, in 5.2.2.2, there is a point that val loss suddenly increases between epoch 2 and 4. Research about the reason will be needed