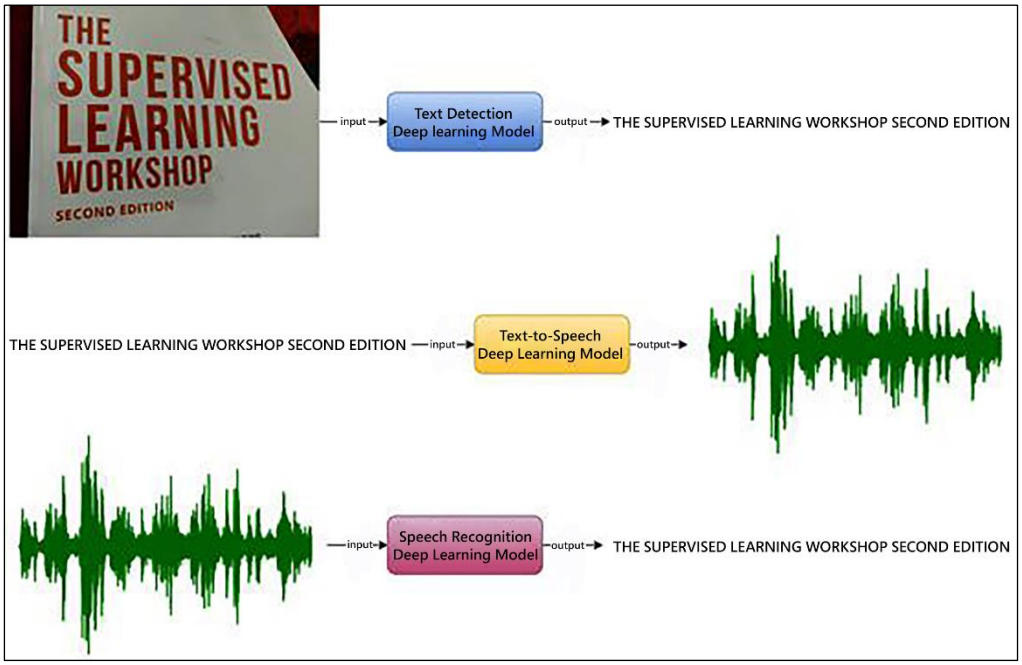
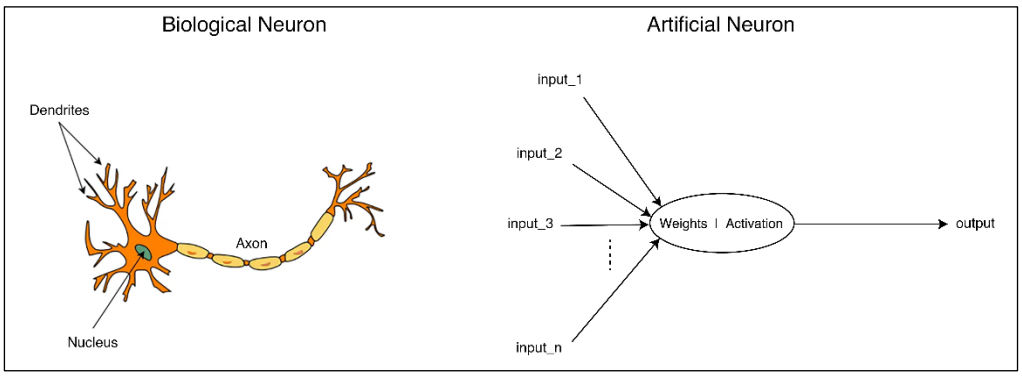
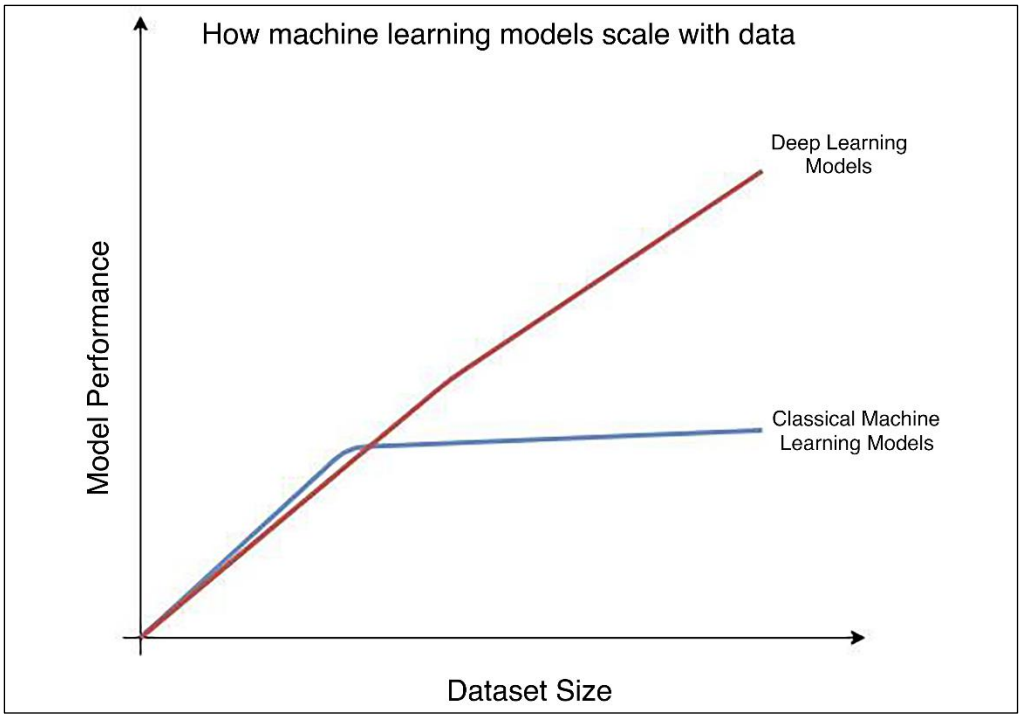
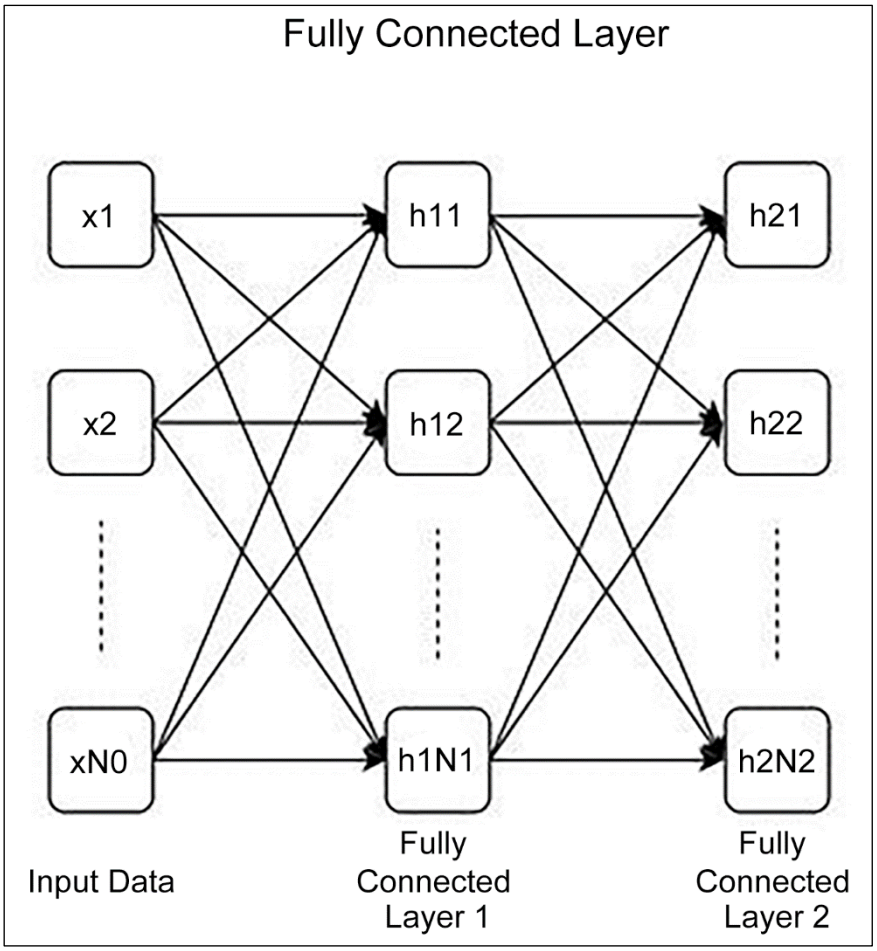
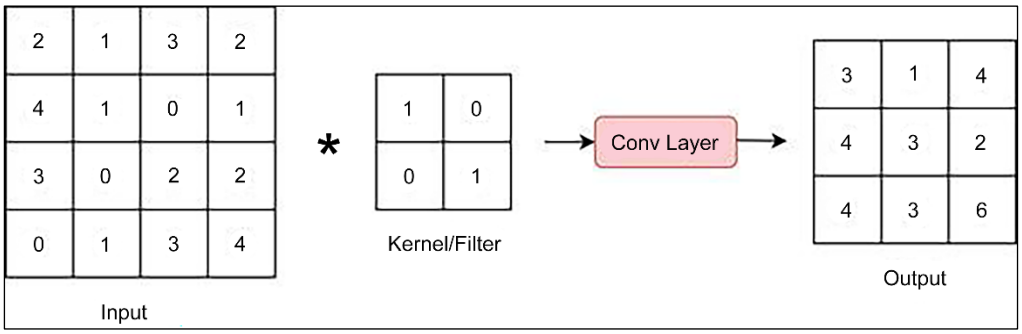
Chapter 1: Overview of Deep Learning Using PyTorch

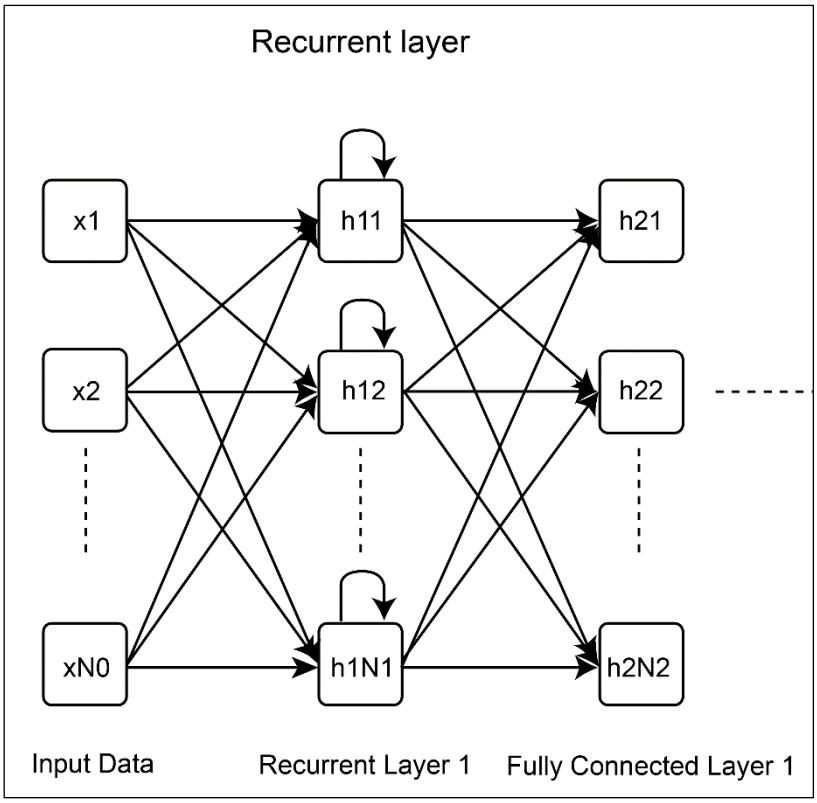


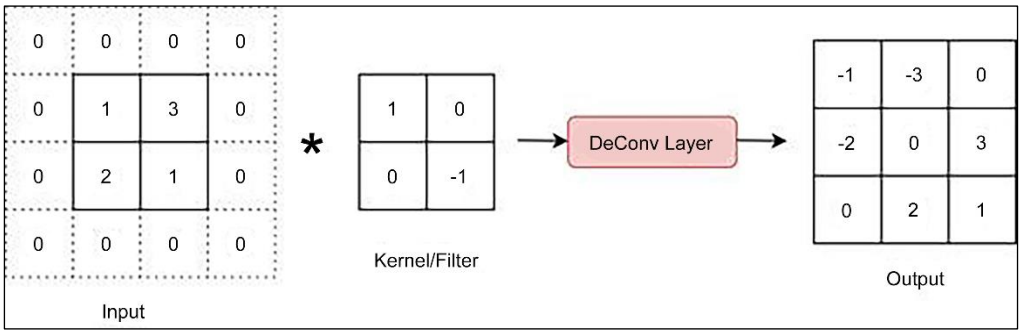


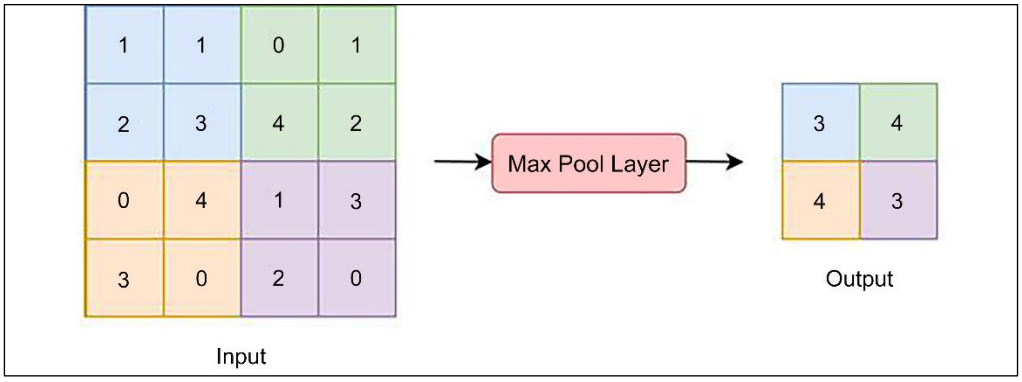

How machine learning models scale with data
Deep Learning Models
Classical Machine Learning Models
Model Performance
Dataset Size

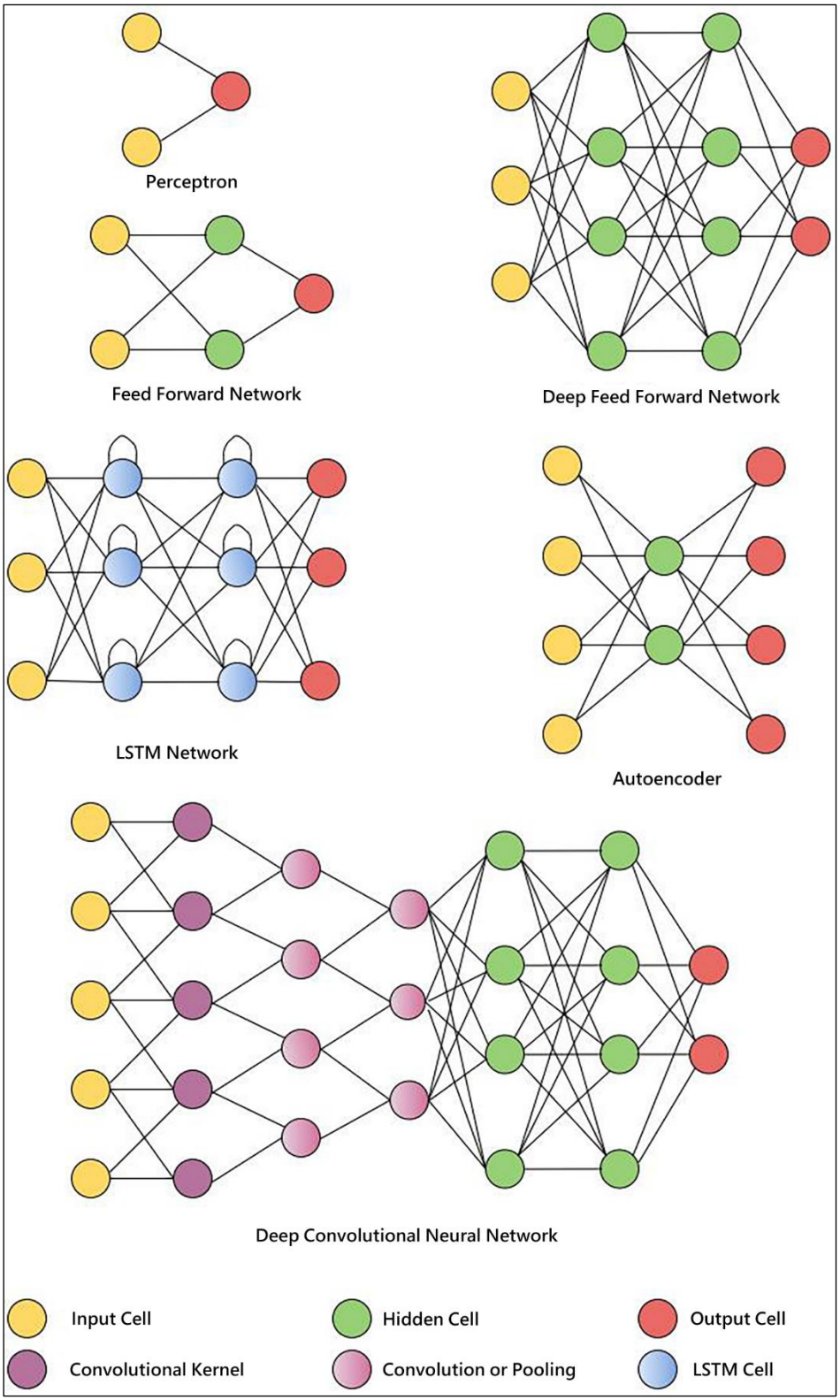

Fully Connected Layer
x1
x2
xN0
h11
h12
h1N1
h21
h22
h2N2
Input Data
Fully Connected Layer 1
Fully Connected Layer 2

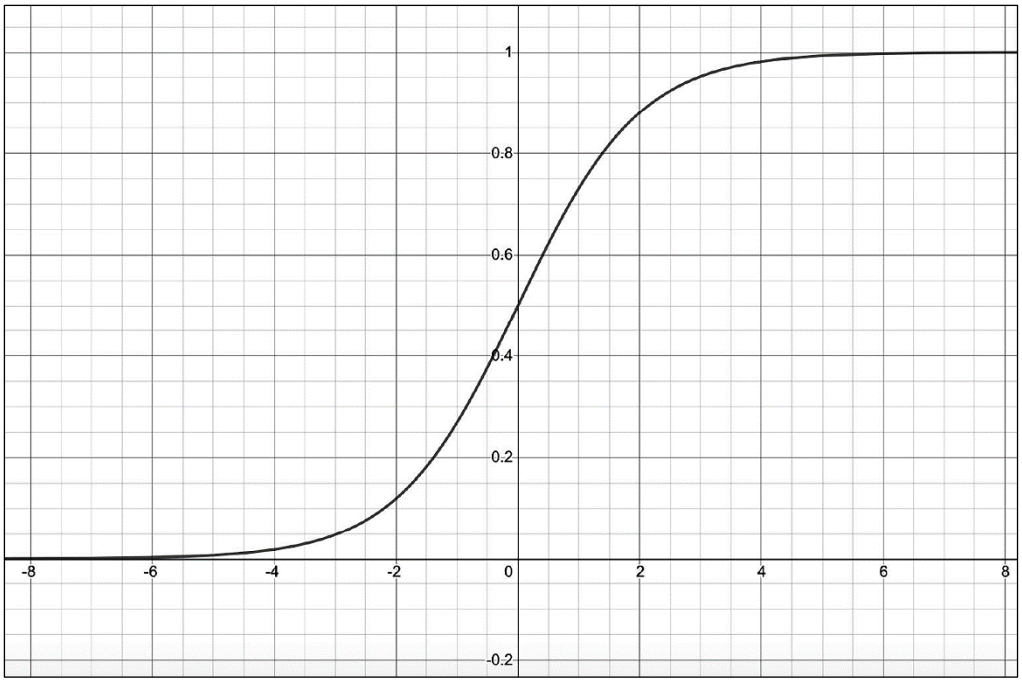

2 1 3 2
4 1 0 1
3 0 2 2
0 1 3 4
Input
*
1 0
0 1
Kernel/Filter
Conv Layer
3 1 4
4 3 2
4 3 6
Output

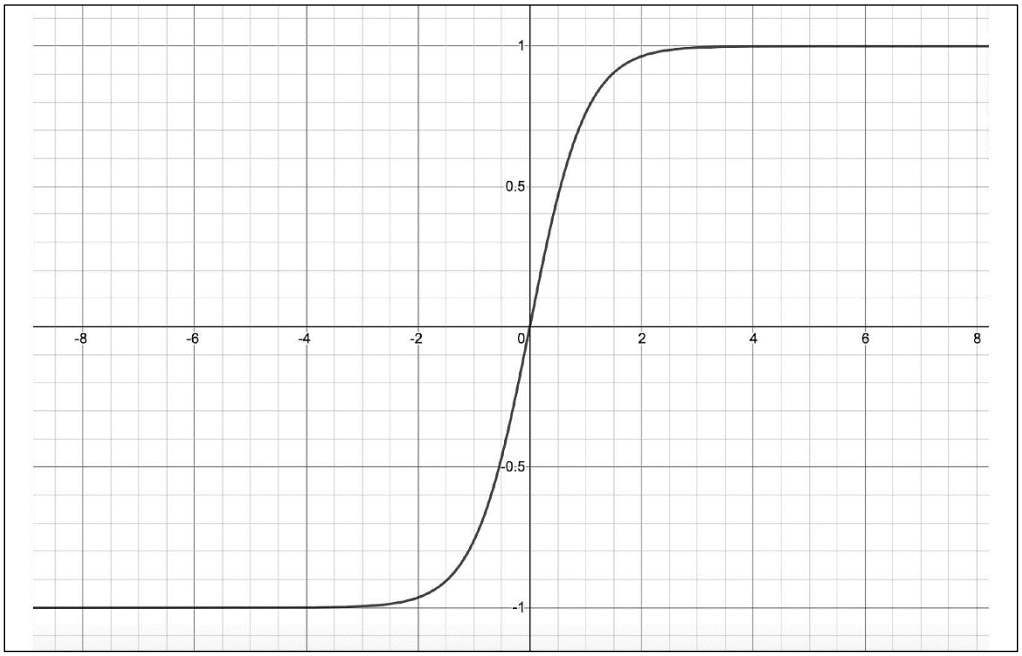

Recurrent layer
x1
x2
xN0
h11
h12
h1N1
h21
h22
h2N2
Input Data
Recurrent Layer 1
Fully Connected Layer 1

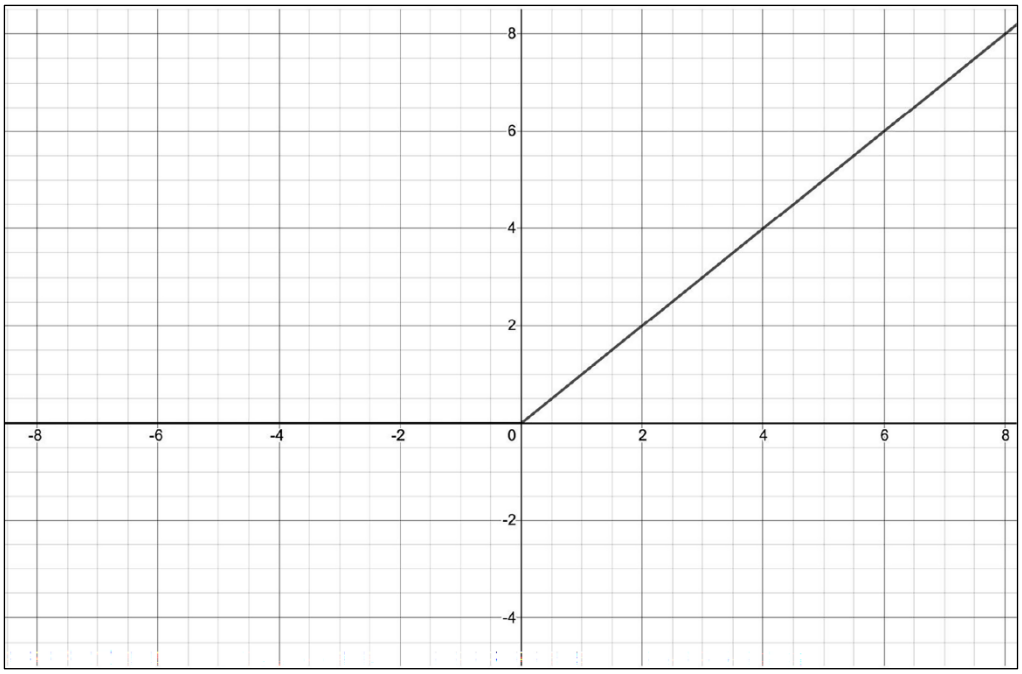

0 0 0 0
0 1 3 0
0 2 1 0
0 0 0 0
Input
*
1 0
0 -1
Kernel/Filter
DeConv Layer
-1 -3 0
-2 0 3
0 2 1
Output

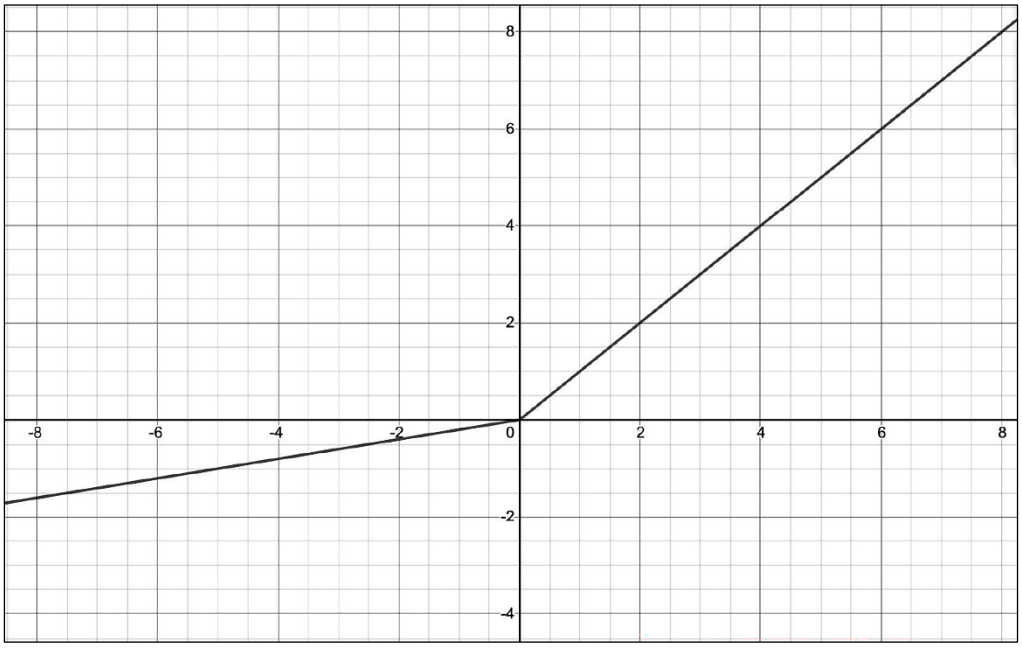

1 1 0 1
2 3 4 2
0 4 1 3
3 0 2 0
Input
Max Pool Layer
3 4
4 3
Output

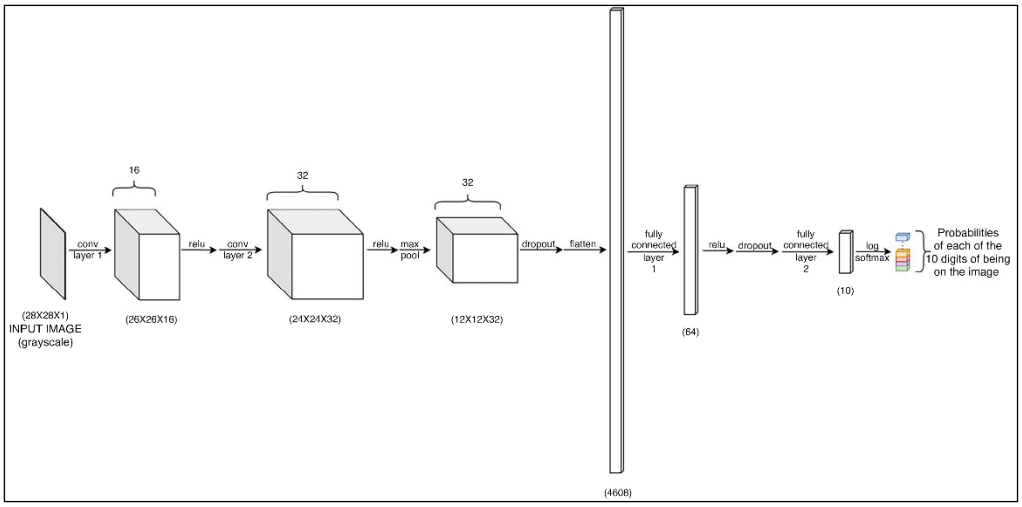

Perceptron
Feed Forward Network
Deep Feed Forward Network
LSTM Network
Autoencoder
Deep Convolutional Neural Network
Input Cell
Hidden Cell
Output Cell
Convolutional Kernel
Convolution or Pooling
LSTM Cell

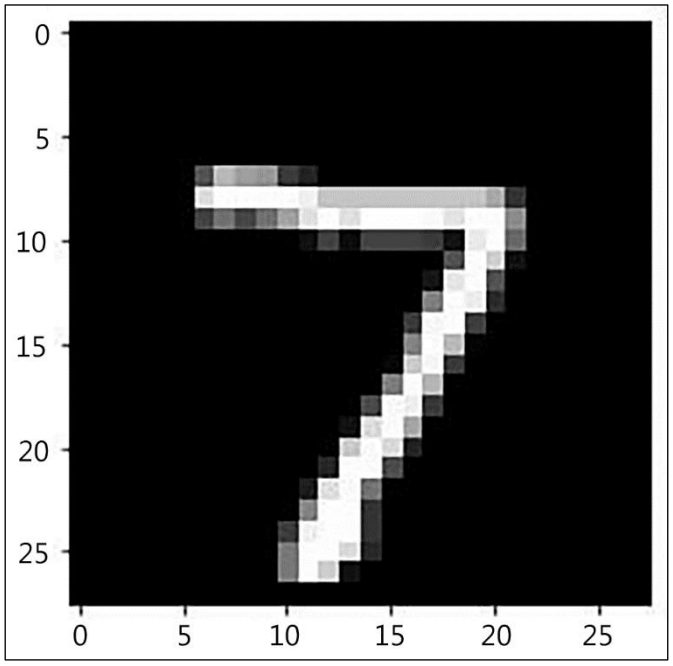

1
0.8
0.6
0.4
0.2
-0.2
-8
-6
-4
-2
0
2
4
6
8


1
0.5
-0.5
-1
-8
-6
-4
-2
0
2
4
6
8

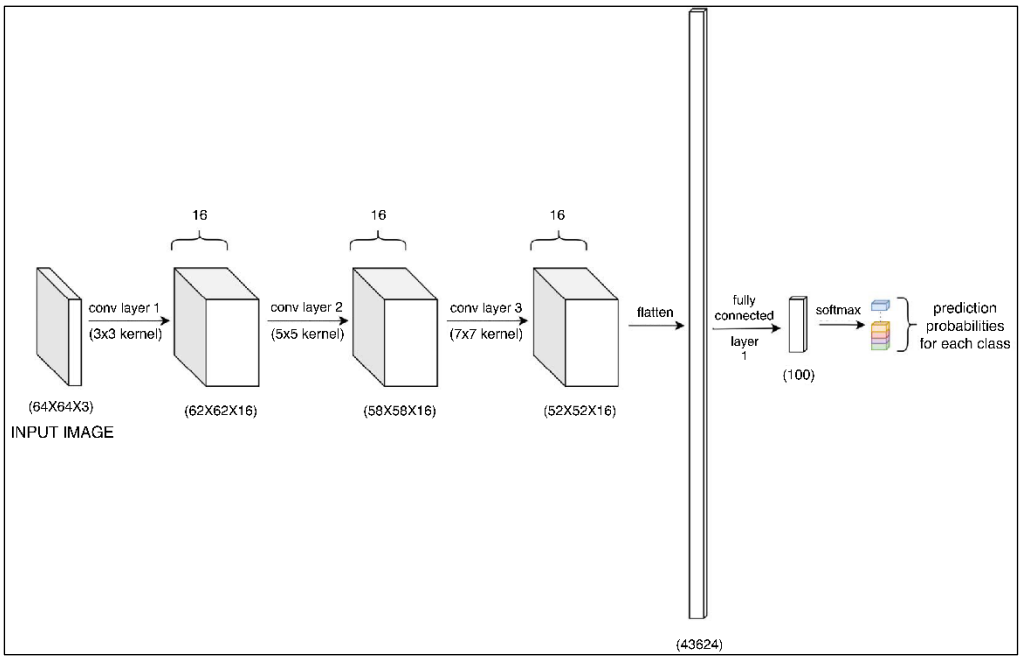

8
6
4
2
-8
-6
-4
-2
0
2
4
6
8
-2
-4

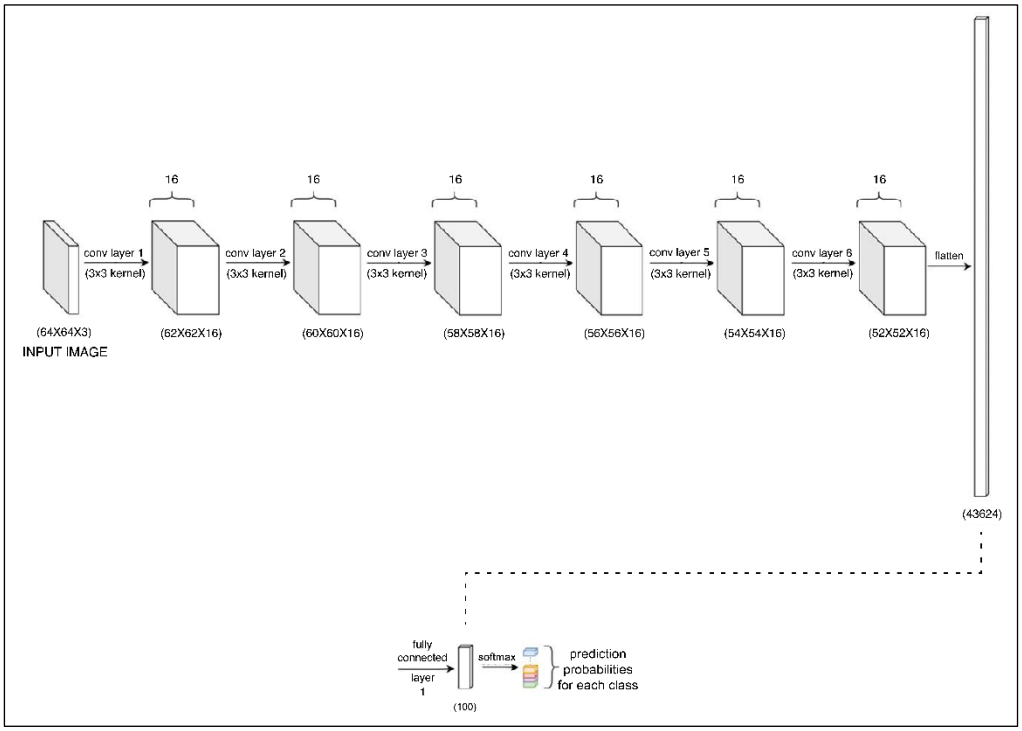

8
6
4
2
-8
-6
-4
-2
0
2
4
6
8
-2
-4

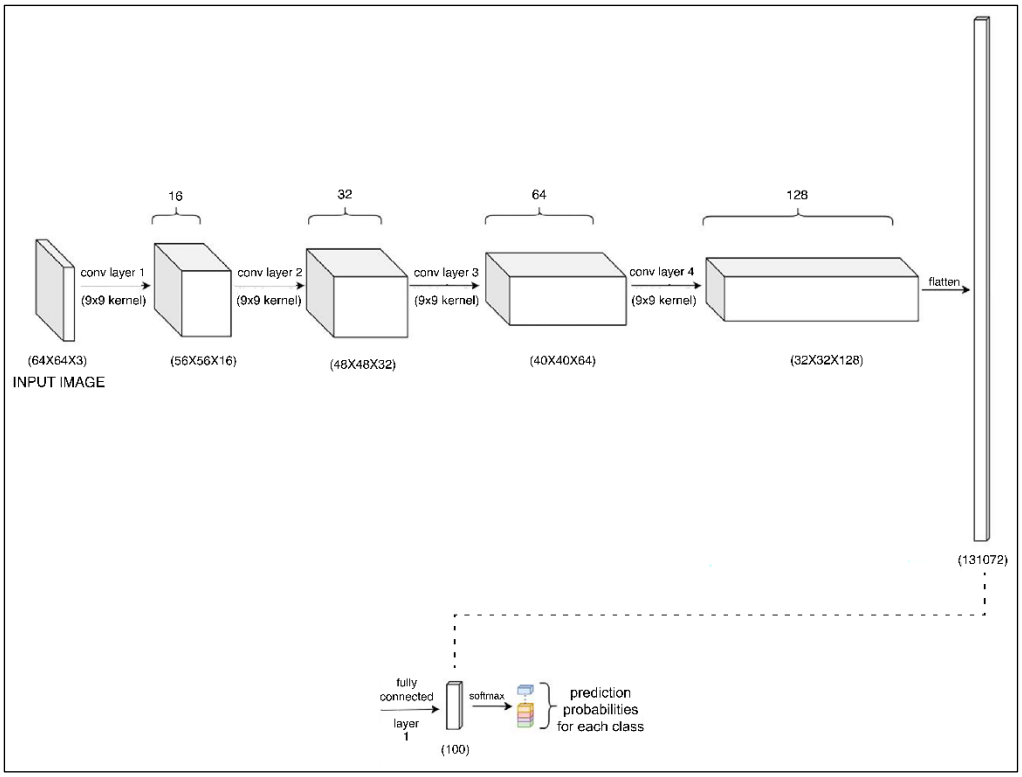

16
32
32
conv layer 1
relu
conv layer 2
relu
max pool
dropout
flatten
fully connected layer 1
relu
dropout
fully connected layer 2
log softmax
Probabilities of each of the 10 digits of being on the image
(28X28X1)
INPUT IMAGE
(grayscale)
(26X26X16)
(24X24X32)
(12X12X32)
(4608)
(64)
(10)

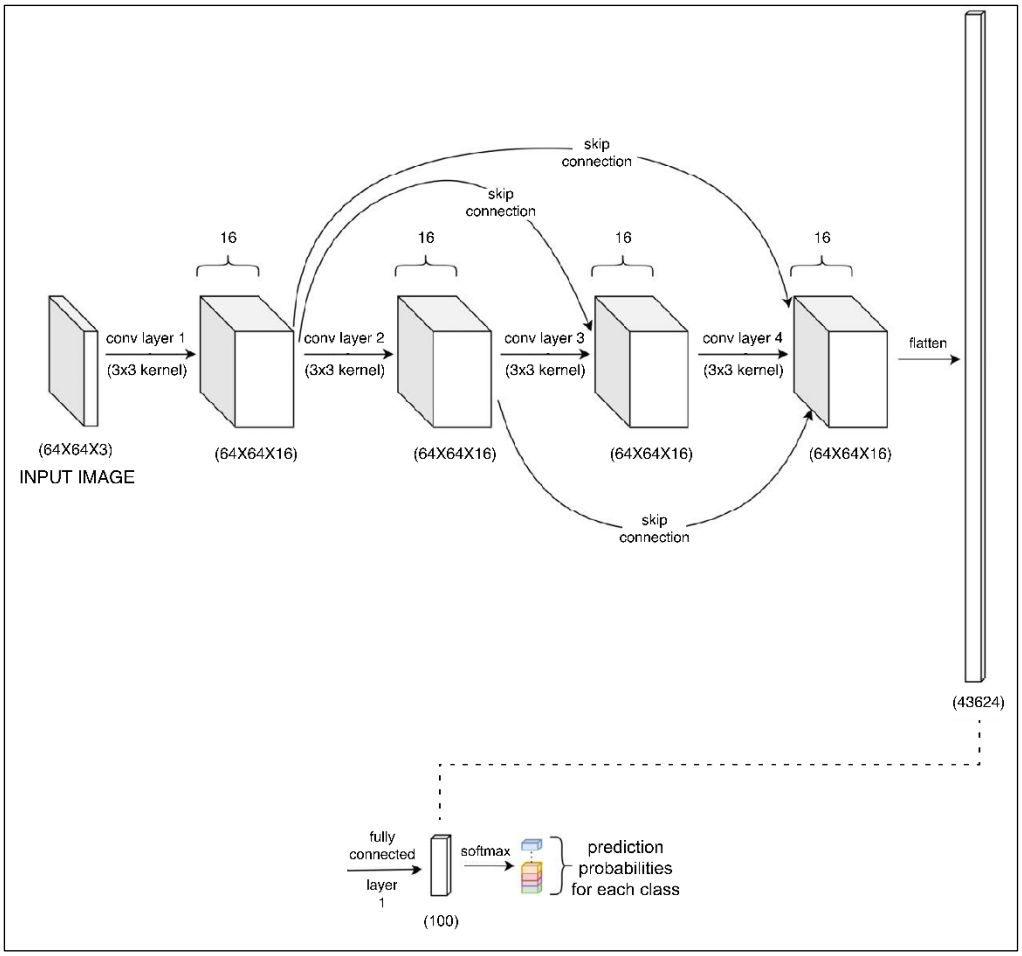

0
5
10
15
20
25
0
5
10
15
20
25

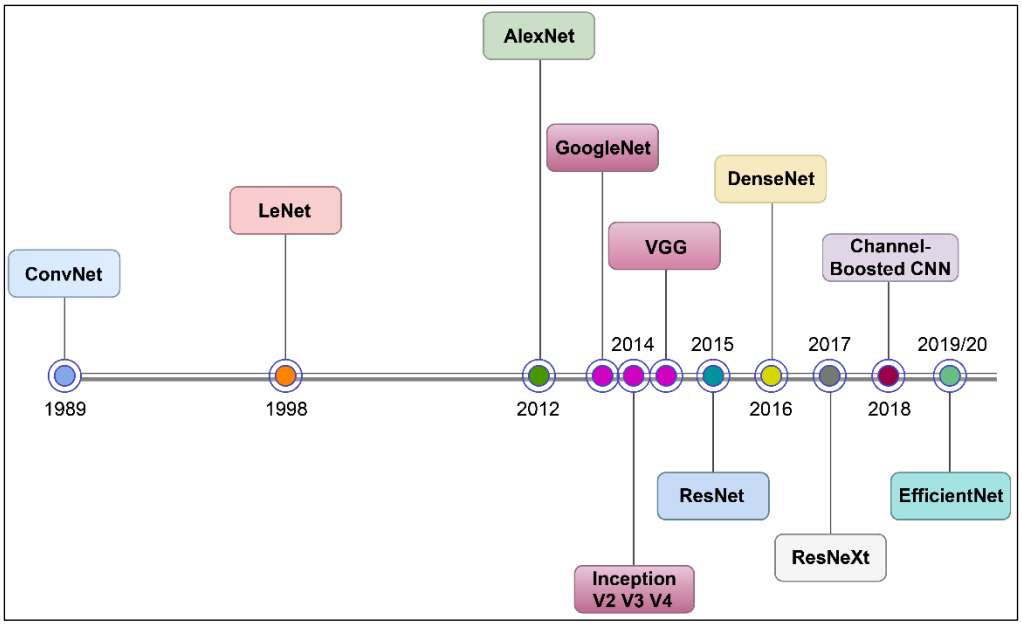
Chapter 2: Deep CNN Architectures

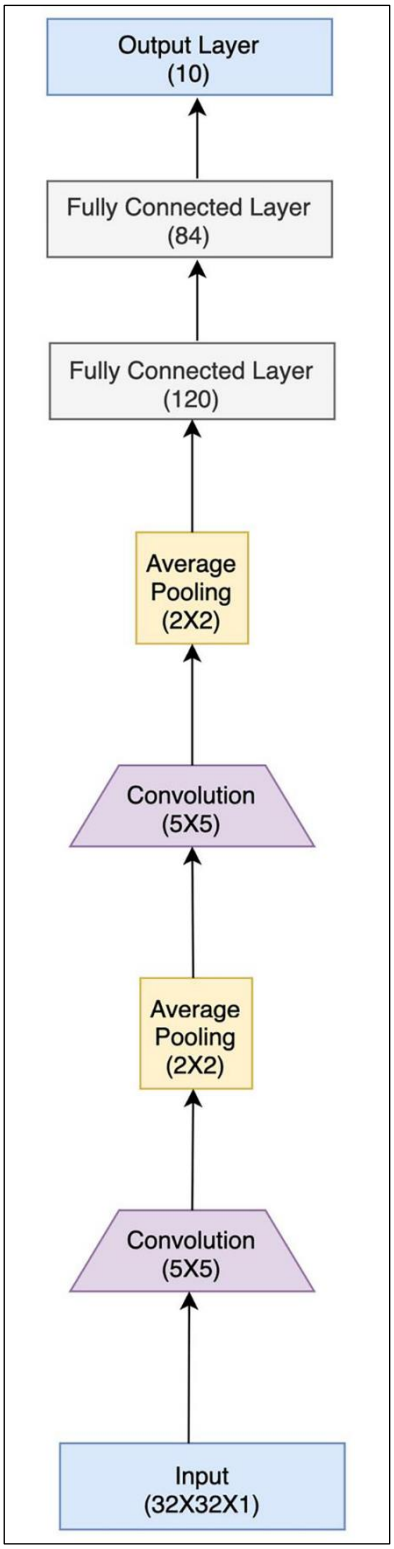


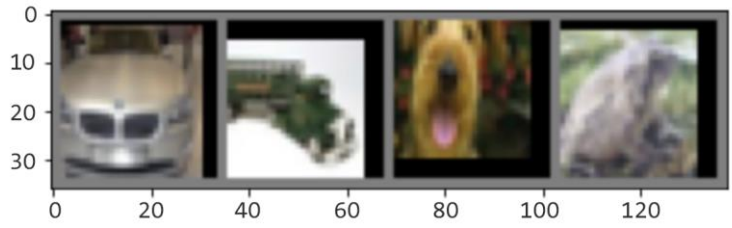



16
32
64
128
conv layer 1
(9x9 kernel)
conv layer 2
(9x9 kernel)
conv layer 3
(9x9 kernel)
conv layer 4
(9x9 kernel)
flatten
(64X64X3)
INPUT IMAGE
(56X56X16)
(48X48X32)
(40X40X64)
(32X32X128)
(131072)
fully
connected
layer
1
(100)
softmax
prediction
probabilities
for each class

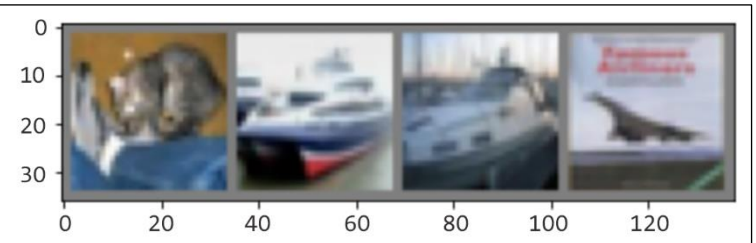

skip
connection
skip
connection
16
16
16
16
conv layer 1
(3x3 kernel)
conv layer 2
(3x3 kernel)
conv layer 3
(3x3 kernel)
conv layer 4
(3x3 kernel)
flatten
(64X64X3)
INPUT IMAGE
(64X64X16)
(64X64X16)
(64X64X16)
(64X64X16)
skip
connection
(43624)
fully
connected
layer
1
(100)
softmax
prediction
probabilities
for each class


AlexNet
GoogleNet
DenseNet
LeNet
VGG
Channel-
Boosted CNN
ConvNet
2014
2015
2017
2019/20
1989
1998
2012
2016
2018
ResNet
EfficientNet
ResNeXt
Inception
V2 V3 V4

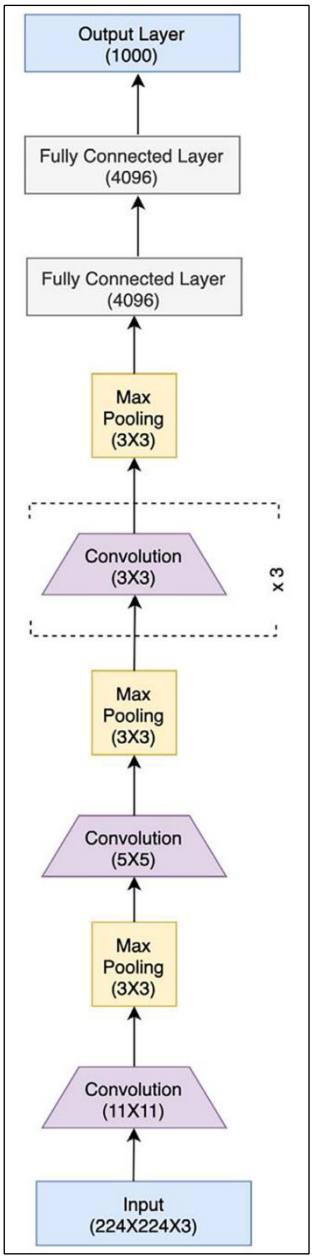

Output Layer
(10)
Fully Connected Layer
(84)
Fully Connected Layer
(120)
Average
Pooling
(2X2)
Convolution
(5X5)
Average
Pooling
(2X2)
Convolution
(5X5)
Input
(32X32X1)

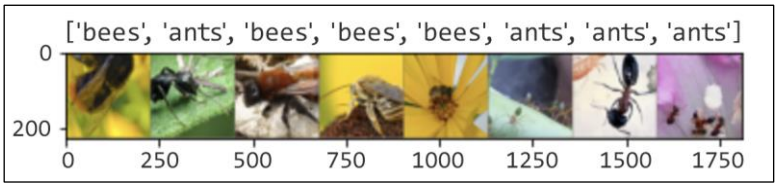



```
car  ||  truck  ||  dog  ||  frog
```




```
Label:          cat  ship  ship plane
Prediction:     cat   car  ship plane
```


Output Layer
(1000)
Fully Connected Layer
(4096)
Fully Connected Layer
(4096)
Max
Pooling
(3X3)
Convolution
(3X3)
x3
Max
Pooling
(3X3)
Convolution
(5X5)
Max
Pooling
(3X3)
Convolution
(11X11)
Input
(224X224X3)


['bees', 'ants', 'bees', 'bees', 'bees', 'ants', 'ants', 'ants']
0
200
0
250
500
750
1000
1250
1500
1750

pred: bees || target: bees



pred: ants || target: ants

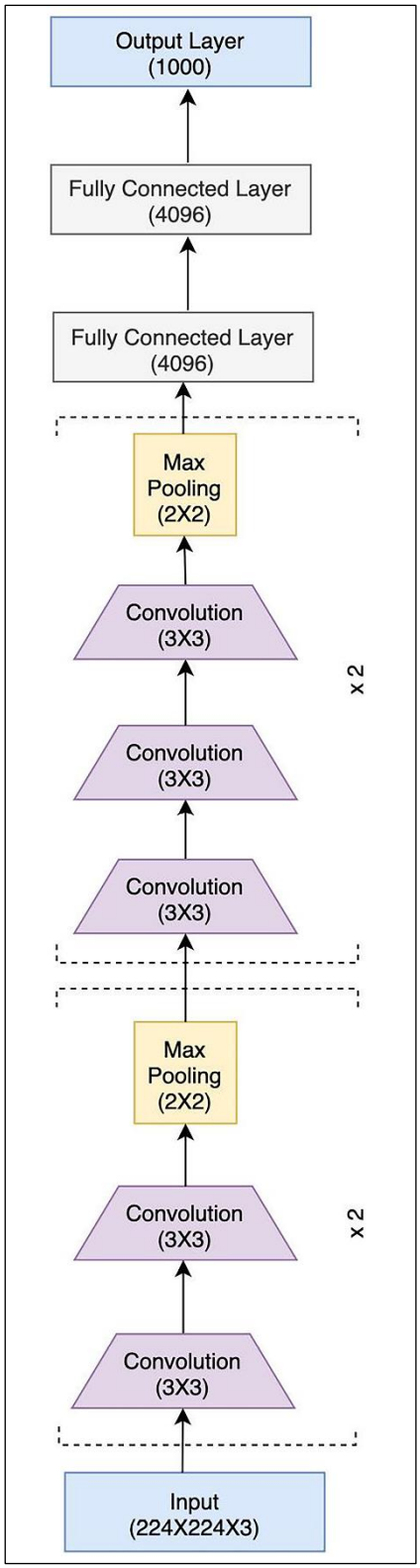


pred: ants || target: ants



pred: bees || target: bees




Output Layer
(1000)
Fully Connected Layer
(4096)
Fully Connected Layer
(4096)
Max
Pooling
(2X2)
Convolution
(3X3)
Convolution
(3X3)
Convolution
(3X3)
x 2
Max
Pooling
(2X2)
Convolution
(3X3)
Convolution
(3X3)
x 2
Input
(224X224X3)

pred: bee



pred: ant, emmet, pismire

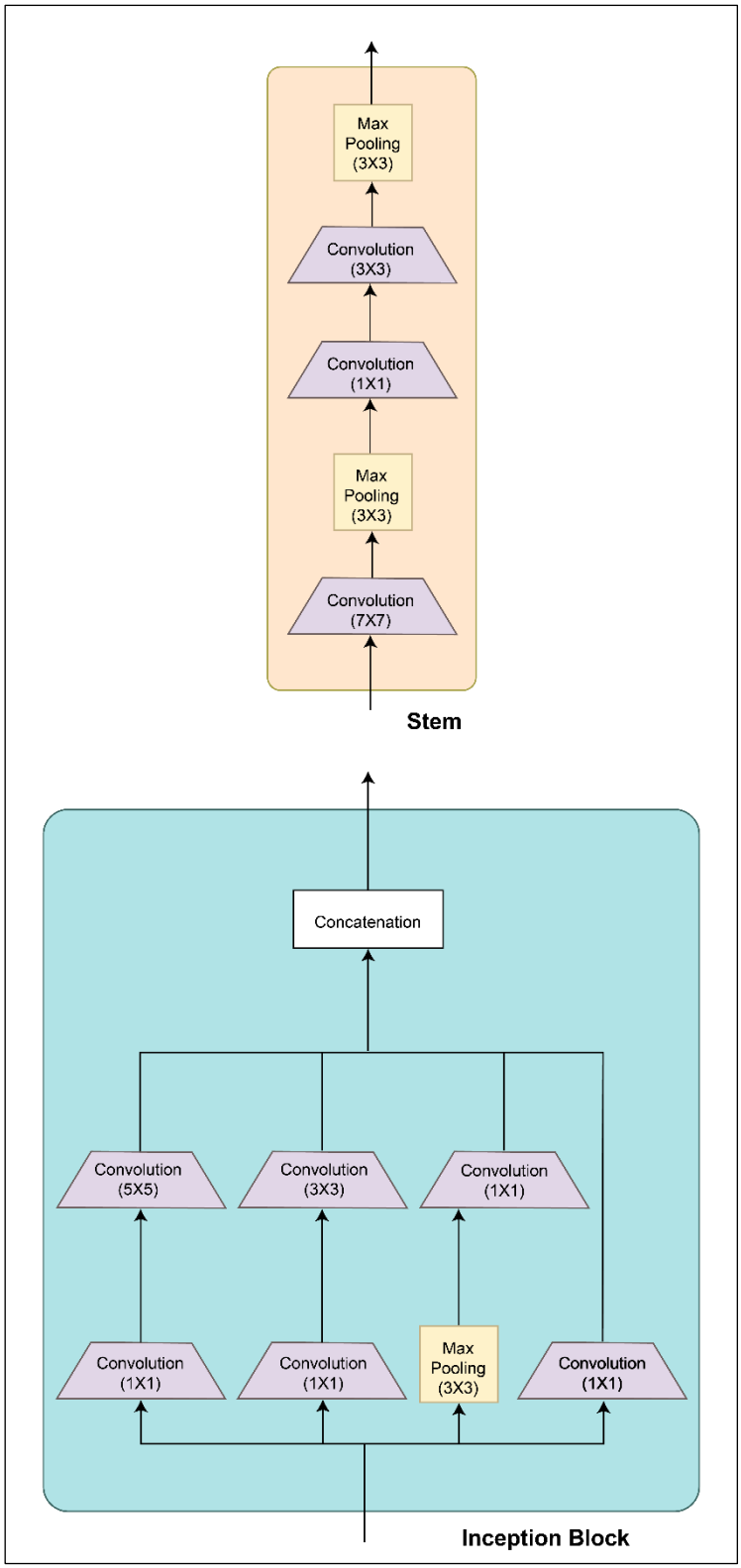


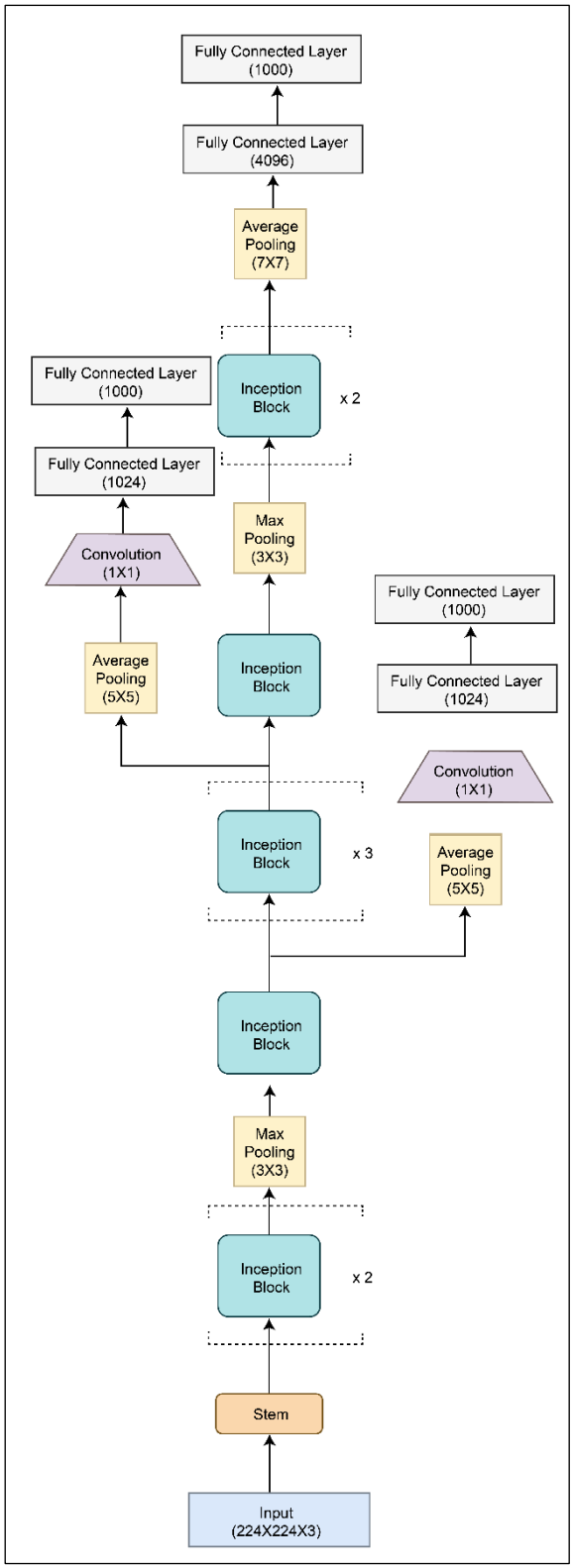
pred: ant, emmet, pismire

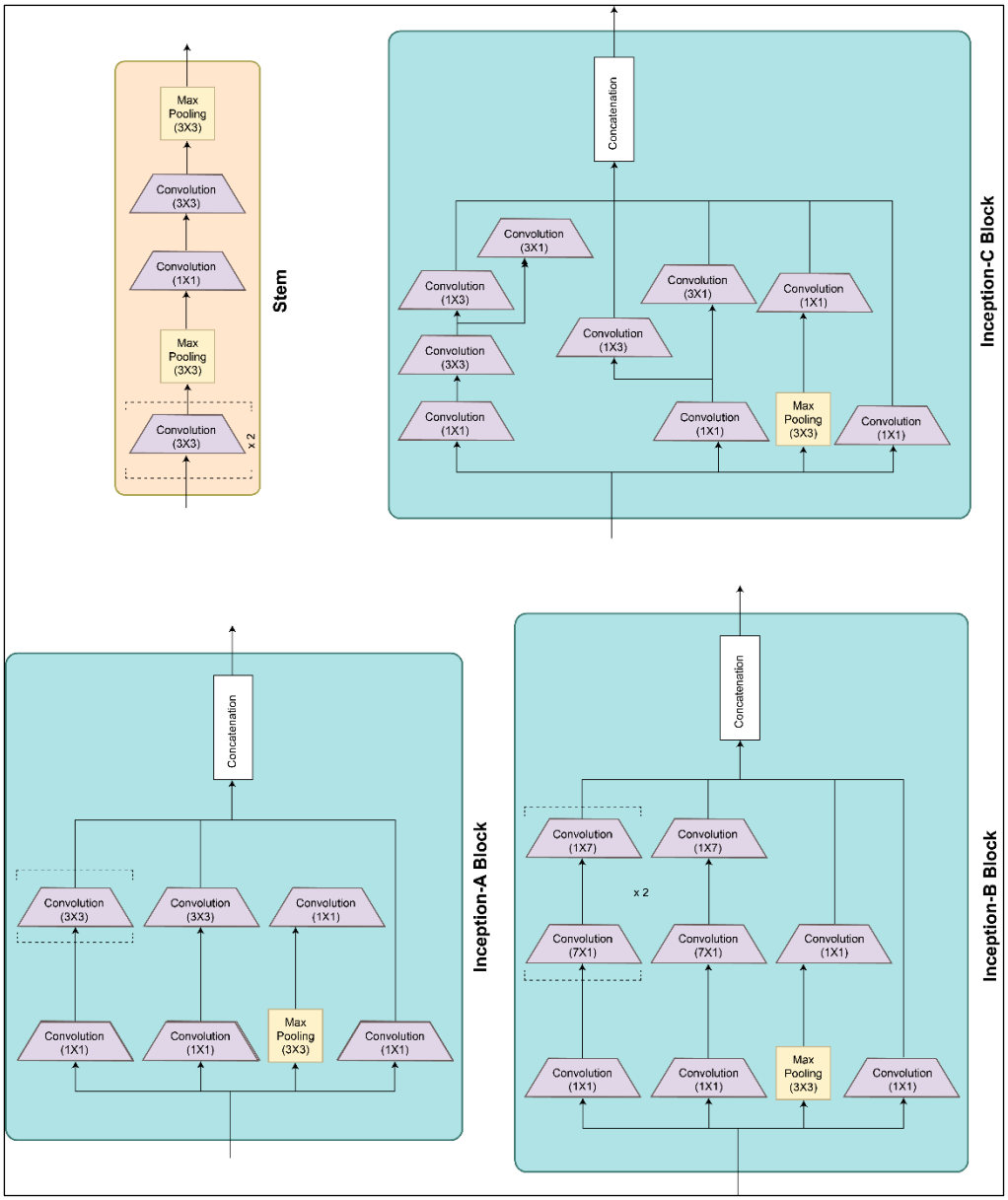


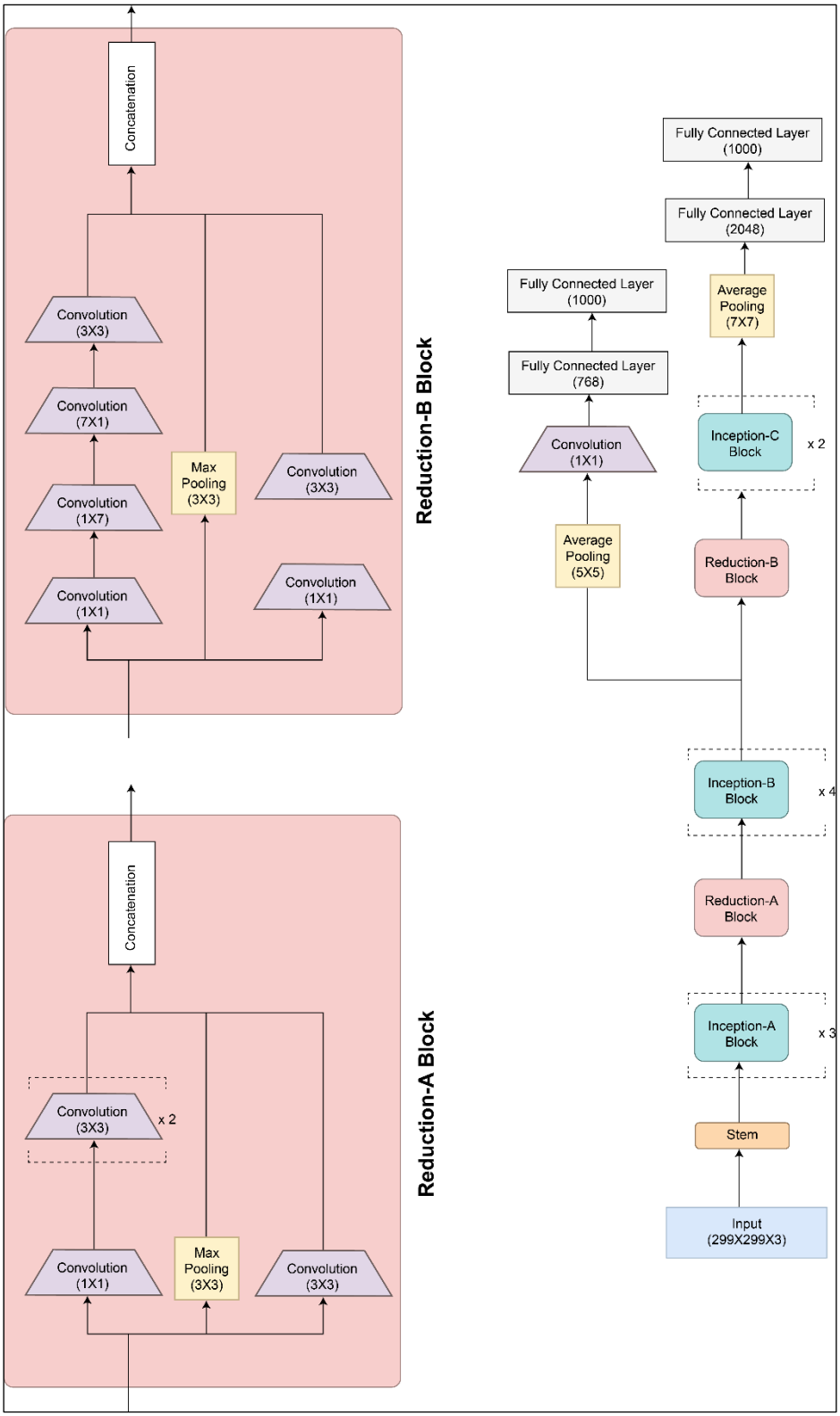
pred: bee




Max Pooling (3X3)
Convolution (3X3)
Convolution (1X1)
Max Pooling (3X3)
Convolution (7X7)
Stem
Concatenation
Convolution (5X5)
Convolution (3X3)
Convolution (1X1)
Convolution (1X1)
Convolution (1X1)
Max Pooling (3X3)
Convolution (1X1)
Inception Block

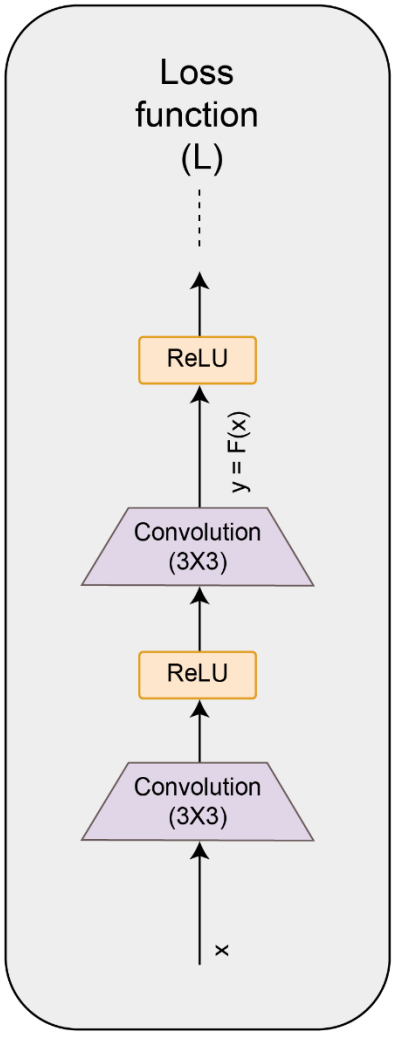

Fully Connected Layer (1000)
Fully Connected Layer (4096)
Average Pooling (7X7)
Inception Block
x 2
Fully Connected Layer (1000)
Fully Connected Layer (1024)
Convolution (1X1)
Max Pooling (3X3)
Average Pooling (5X5)
Inception Block
Fully Connected Layer (1000)
Fully Connected Layer (1024)
Convolution (1X1)
Inception Block
x 3
Average Pooling (5X5)
Inception Block
Max Pooling (3X3)
Inception Block
x 2
Stem
Input (224X224X3)


Stem
Max Pooling (3X3)
Convolution (3X3)
Convolution (1X1)
Max Pooling (3X3)
Convolution (3X3)
x 2
Inception-C Block
Concatenation
Convolution (3X1)
Convolution (1X3)
Convolution (3X1)
Convolution (1X1)
Convolution (1X3)
Convolution (3X3)
Convolution (1X1)
Convolution (1X1)
Max Pooling (3X3)
Convolution (1X1)
Inception-A Block
Concatenation
Convolution (3X3)
Convolution (3X3)
Convolution (1X1)
Convolution (1X1)
Convolution (1X1)
Max Pooling (3X3)
Convolution (1X1)
Inception-B Block
Concatenation
Convolution (1X7)
Convolution (1X7)
x 2
Convolution (7X1)
Convolution (7X1)
Convolution (1X1)
Convolution (1X1)
Convolution (1X1)
Max Pooling (3X3)
Convolution (1X1)

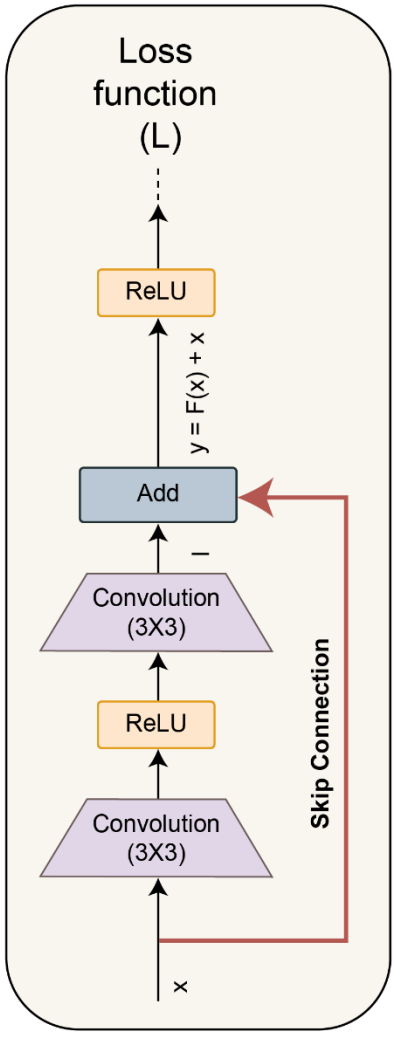

Concatenation
Convolution (3X3)
Convolution (7X1)
Convolution (1X7)
Convolution (1X1)
Max Pooling (3X3)
Convolution (3X3)
Convolution (1X1)
Reduction-B Block
Concatenation
Convolution (3X3)
x 2
Convolution (1X1)
Max Pooling (3X3)
Convolution (3X3)
Reduction-A Block
Fully Connected Layer (1000)
Fully Connected Layer (2048)
Fully Connected Layer (1000)
Average Pooling (7X7)
Fully Connected Layer (768)
Convolution (1X1)
Inception-C Block
x 2
Average Pooling (5X5)
Reduction-B Block
Inception-B Block
x 4
Reduction-A Block
Inception-A Block
x 3
Stem
Input (299X299X3)

$$\frac{\delta L}{\delta x} = \frac{\delta L}{\delta y} * \frac{\delta y}{\delta x}$$
$$= \frac{\delta L}{\delta y} * F'(x)$$

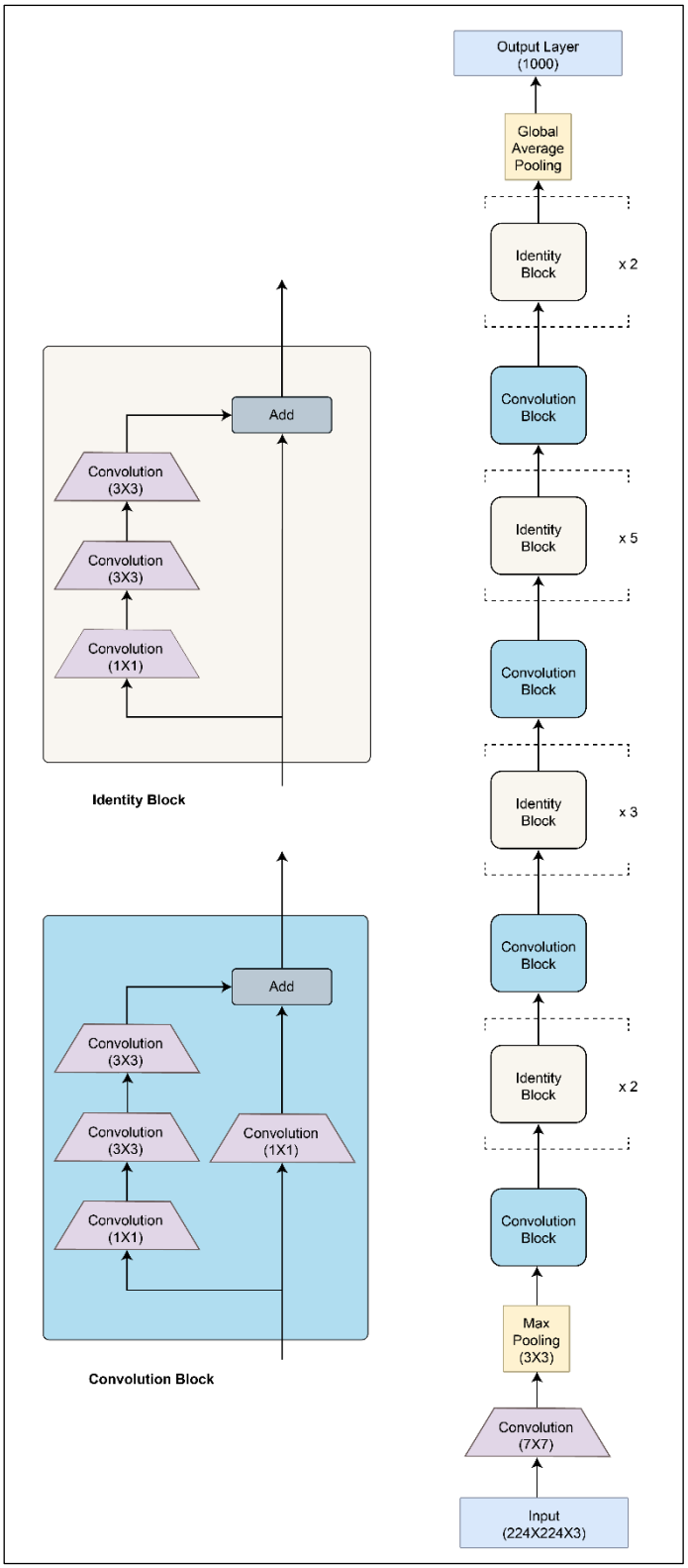
$$\frac{\delta L}{\delta x} = \frac{\delta L}{\delta y} * \frac{\delta y}{\delta x}$$
$$= \frac{\delta L}{\delta y} * F'(x) + \underbrace{\frac{\delta L}{\delta y}}_{\text{helps stabilize gradients}}$$

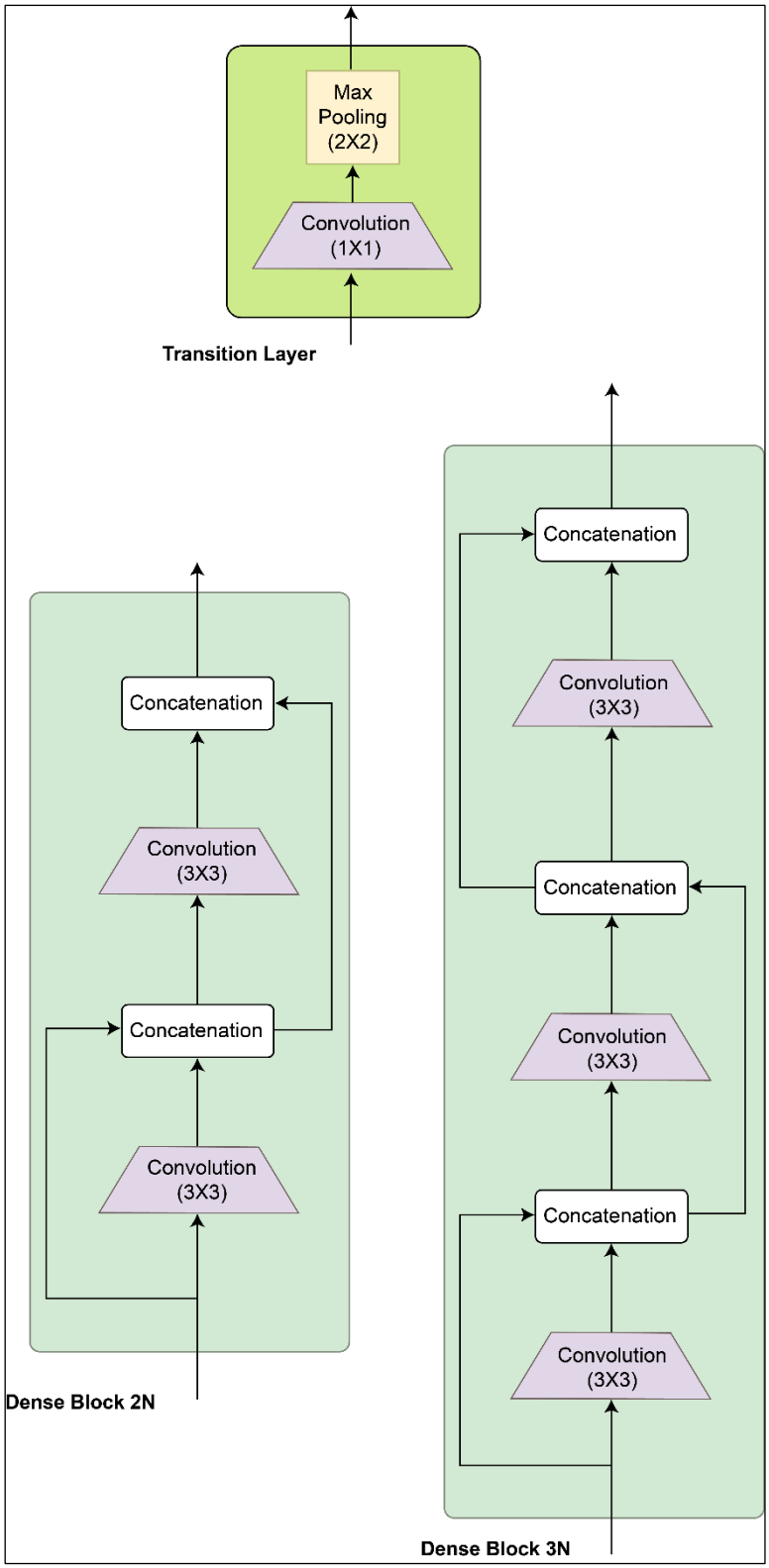


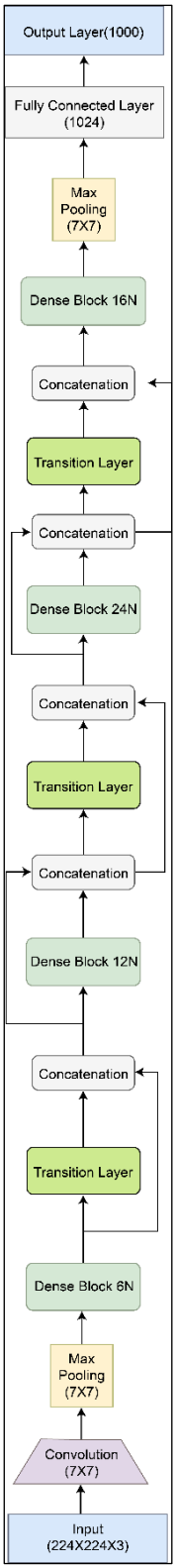
Regular Convolutional Block
(No skip connection)

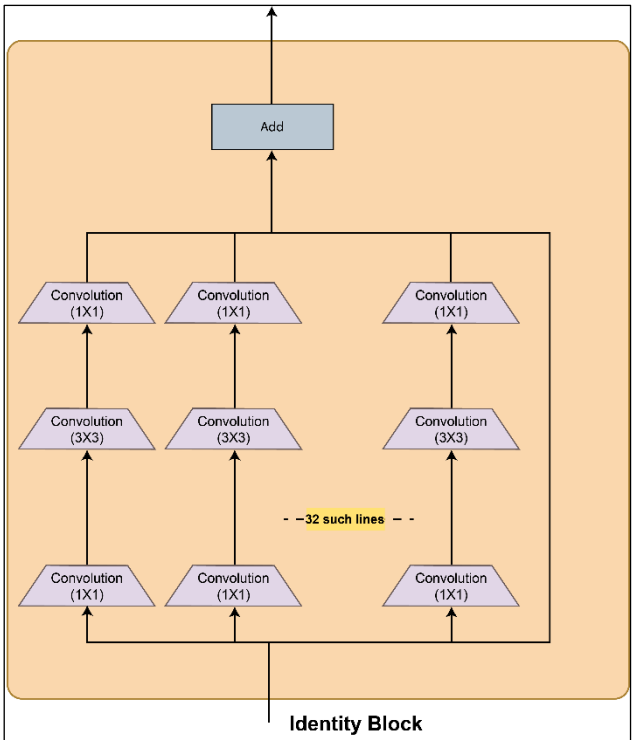


Residual Convolutional Block
(skip connection)

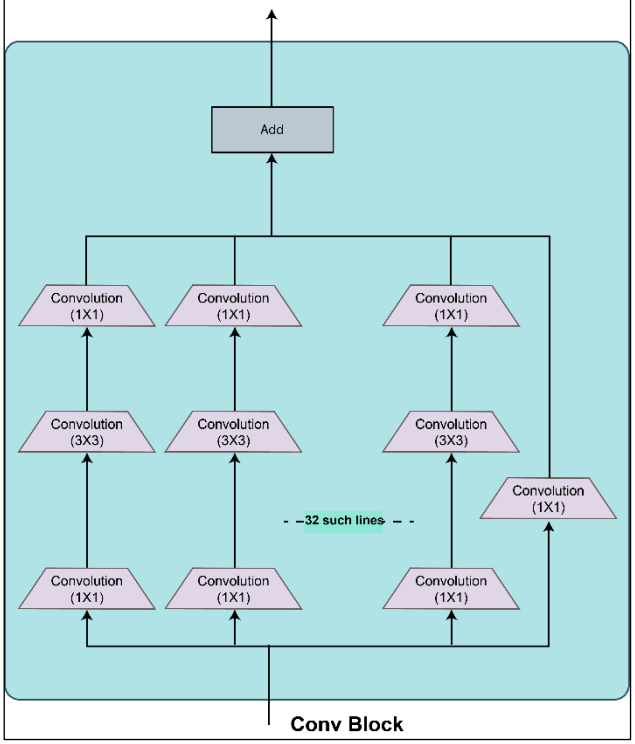

Output Layer
(1000)
Global
Average
Pooling
Identity
Block
x 2
Convolution
Block
Identity
Block
x 5
Convolution
Block
Identity
Block
x 3
Convolution
Block
Identity
Block
x 2
Convolution
Block
Max
Pooling
(3X3)
Convolution
(7X7)
Input
(224X224X3)
Add
Convolution
(3X3)
Convolution
(3X3)
Convolution
(1X1)
Identity Block
Add
Convolution
(3X3)
Convolution
(3X3)
Convolution
(1X1)
Convolution
(1X1)
Convolution Block


Max Pooling (2X2)
Convolution (1X1)
Transition Layer
Concatenation
Convolution (3X3)
Concatenation
Convolution (3X3)
Dense Block 2N
Concatenation
Convolution (3X3)
Concatenation
Convolution (3X3)
Concatenation
Convolution (3X3)
Dense Block 3N

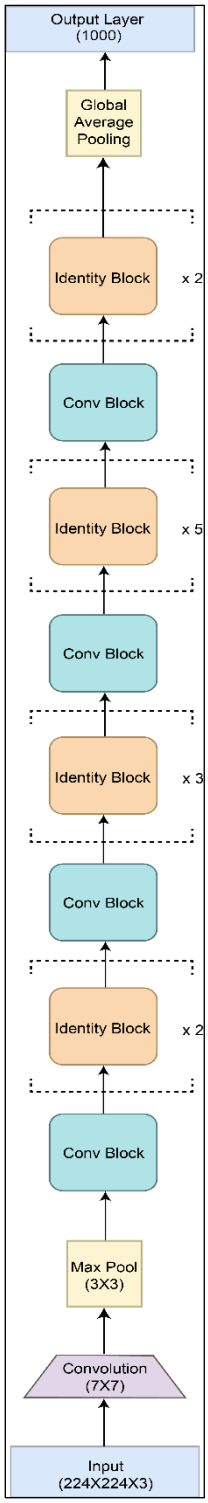

Output Layer(1000)
Fully Connected Layer (1024)
Max Pooling (7X7)
Dense Block 16N
Concatenation
Transition Layer
Concatenation
Dense Block 24N
Concatenation
Transition Layer
Concatenation
Dense Block 12N
Concatenation
Transition Layer
Dense Block 6N
Max Pooling (7X7)
Convolution (7X7)
Input (224X224X3)



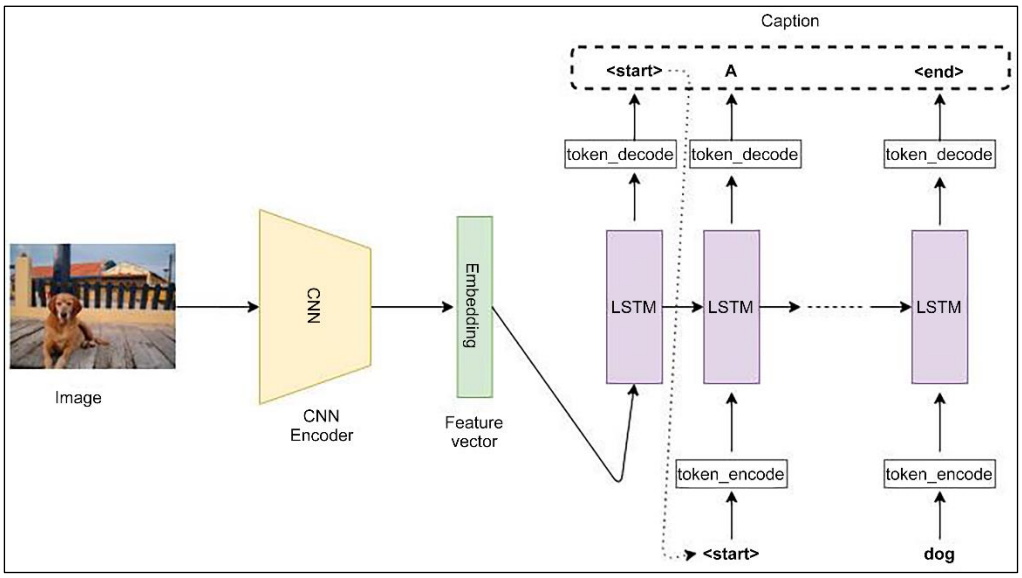
Identity Block

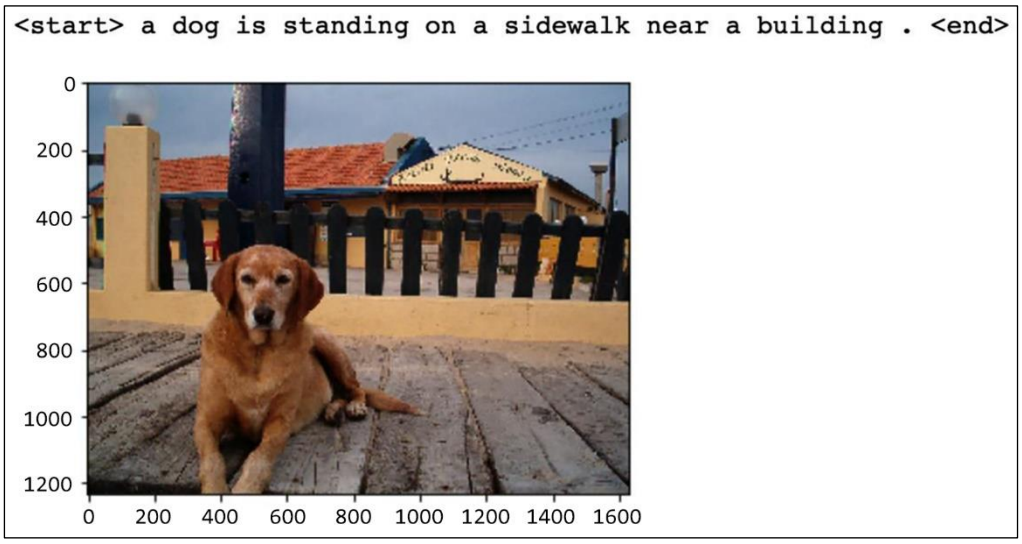


Conv Block

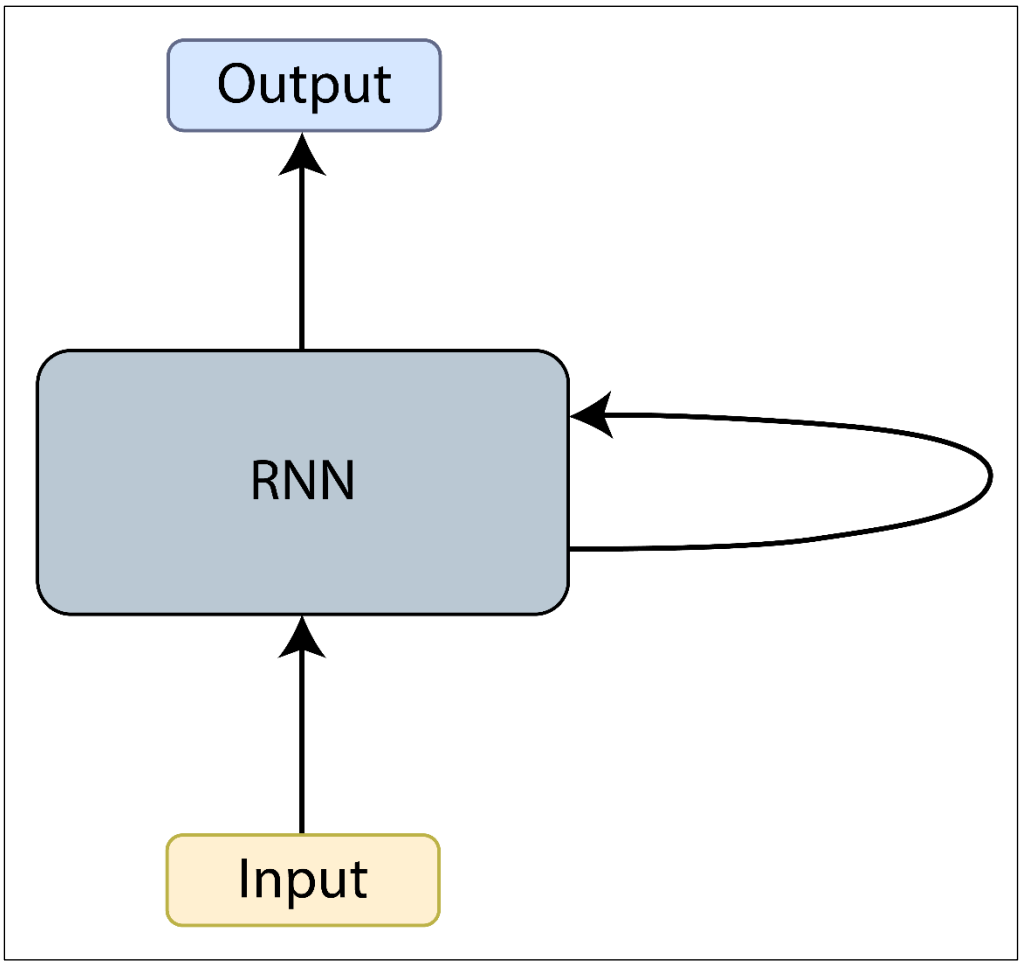

Output Layer
(1000)
Global
Average
Pooling
Identity Block
x 2
Conv Block
Identity Block
x 5
Conv Block
Identity Block
x 3
Conv Block
Identity Block
x 2
Conv Block
Max Pool
(3X3)
Convolution
(7X7)
Input
(224X224X3)

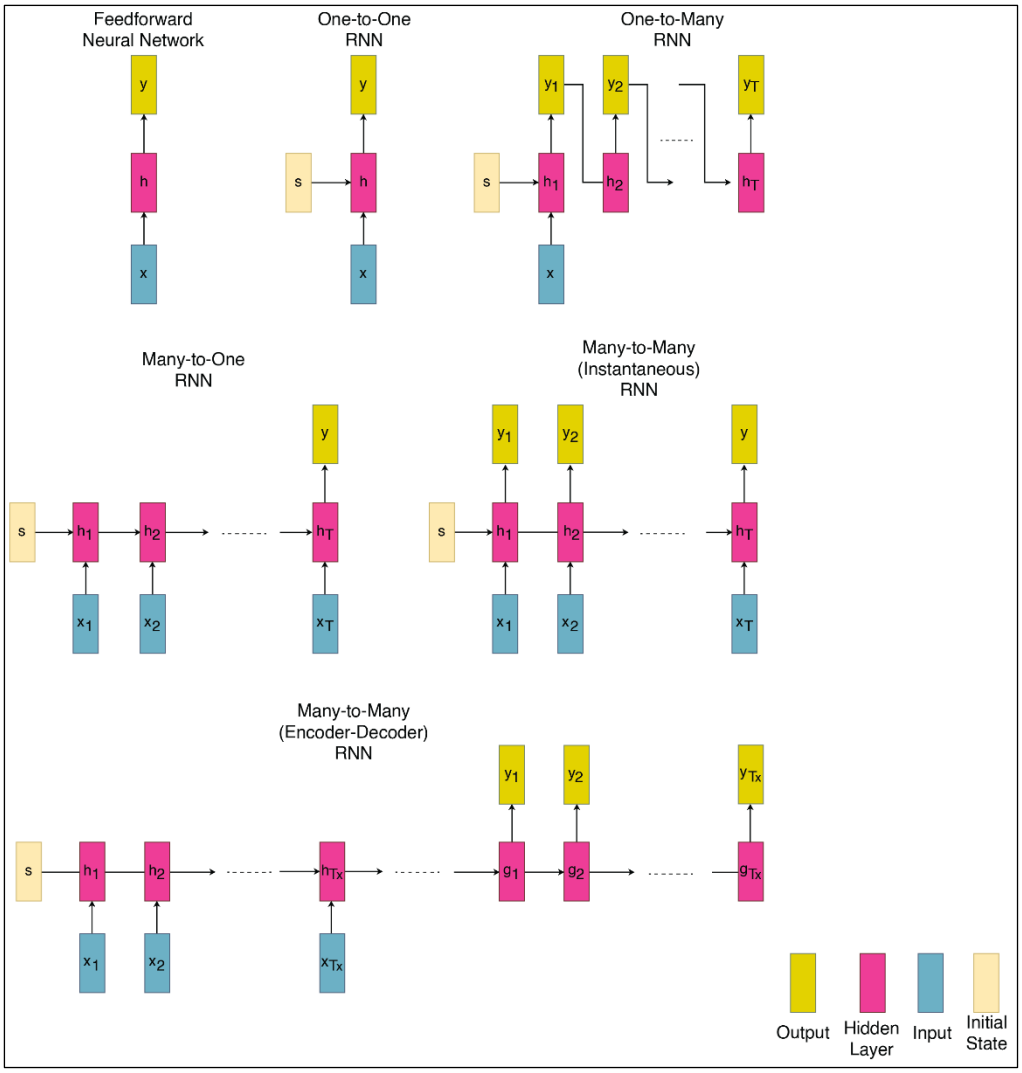
Chapter 3: Combining CNNs and LSTMs

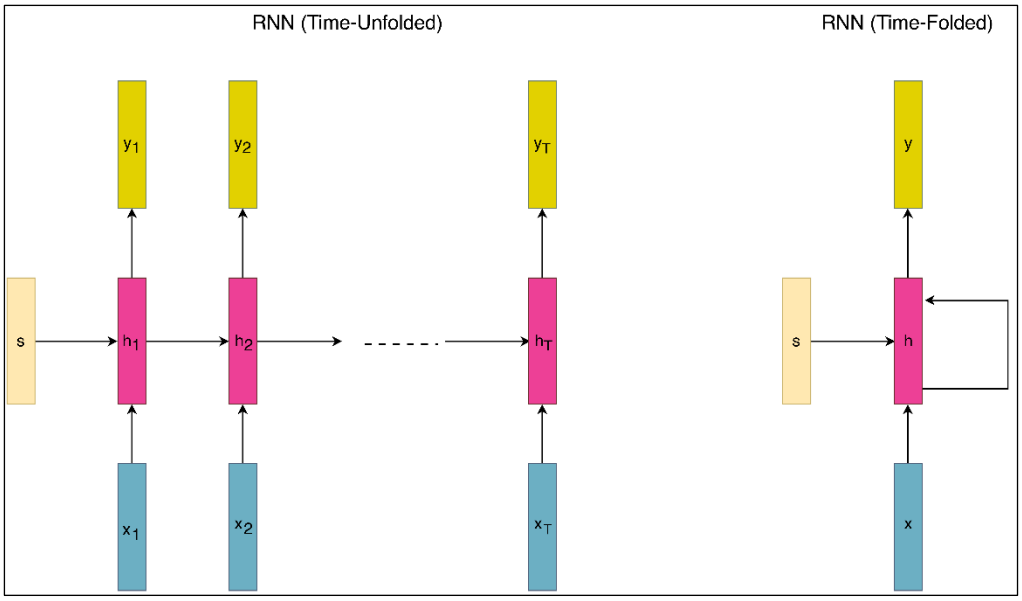


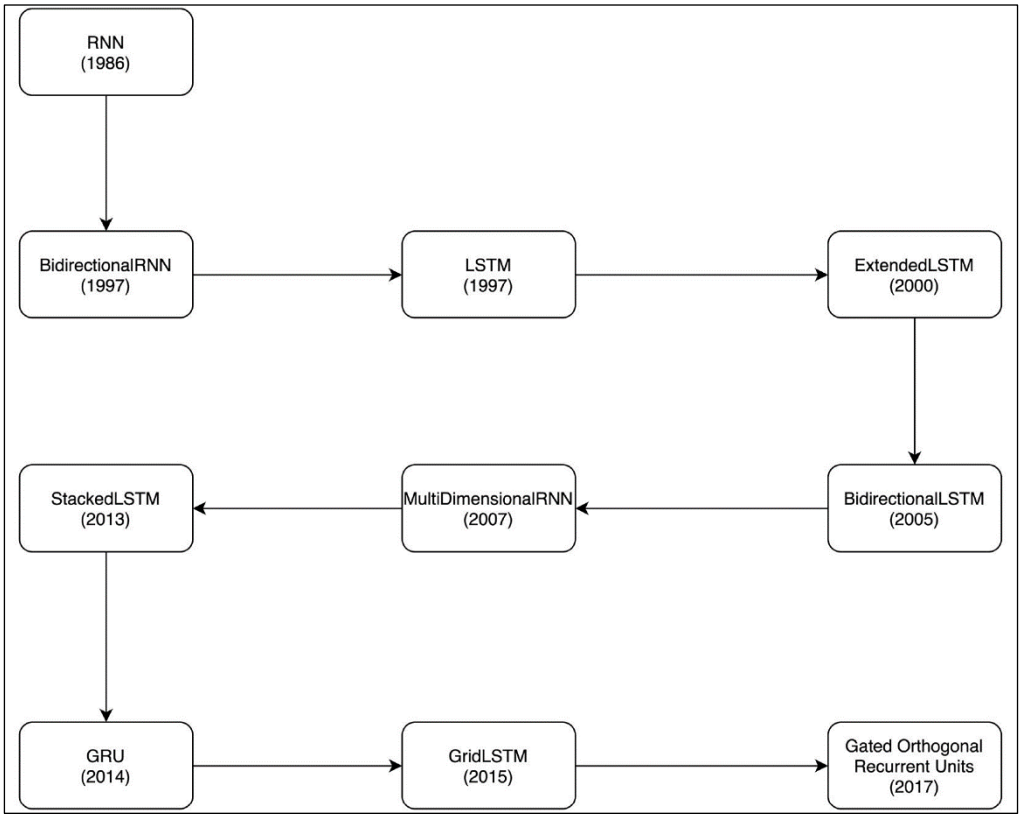


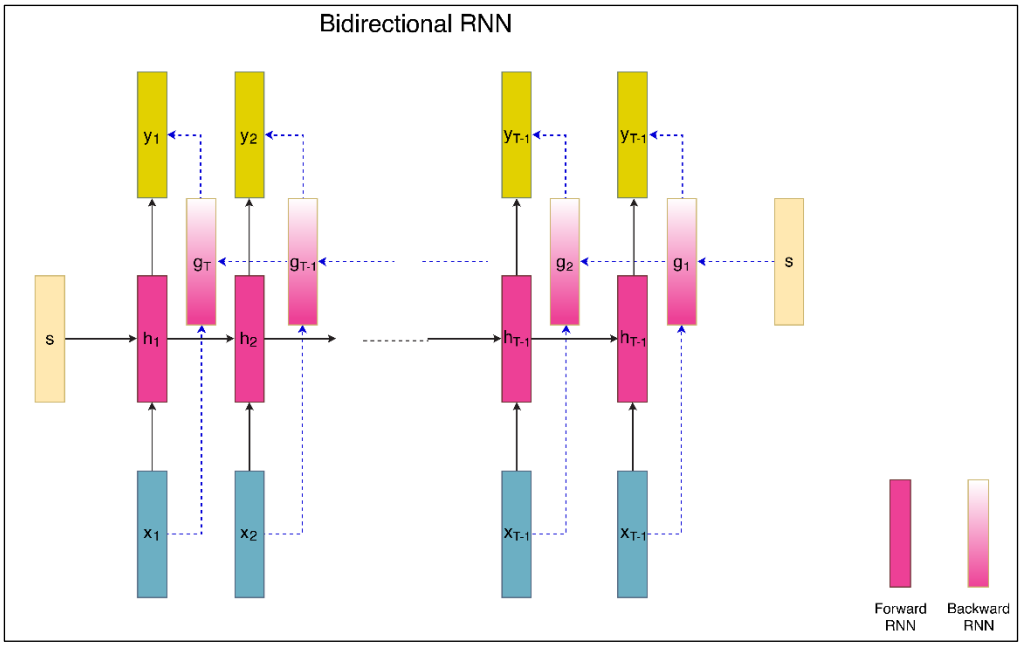
Chapter 4: Deep Recurrent Model Architectures

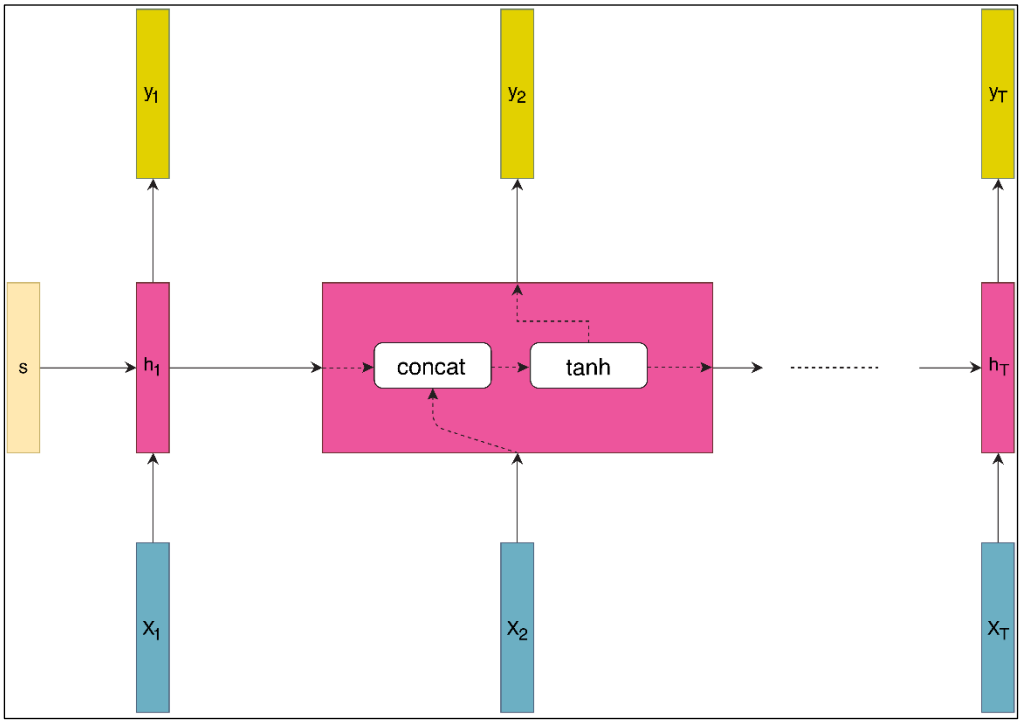


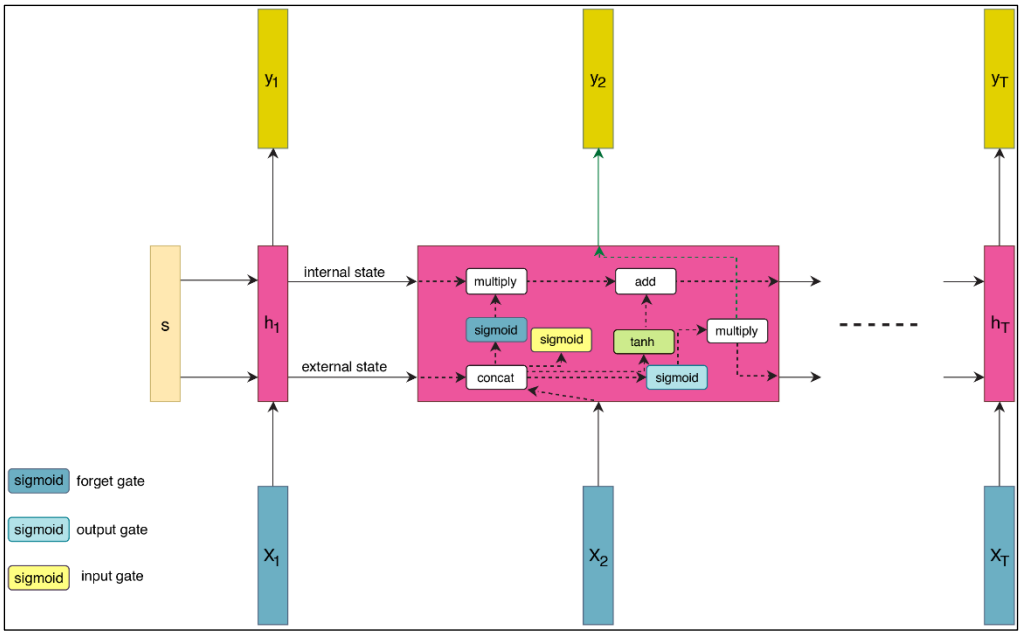

Feedforward Neural Network
One-to-One RNN
One-to-Many RNN
Many-to-One RNN
Many-to-Many (Instantaneous) RNN
Many-to-Many (Encoder-Decoder) RNN
Output
Hidden Layer
Input
Initial State

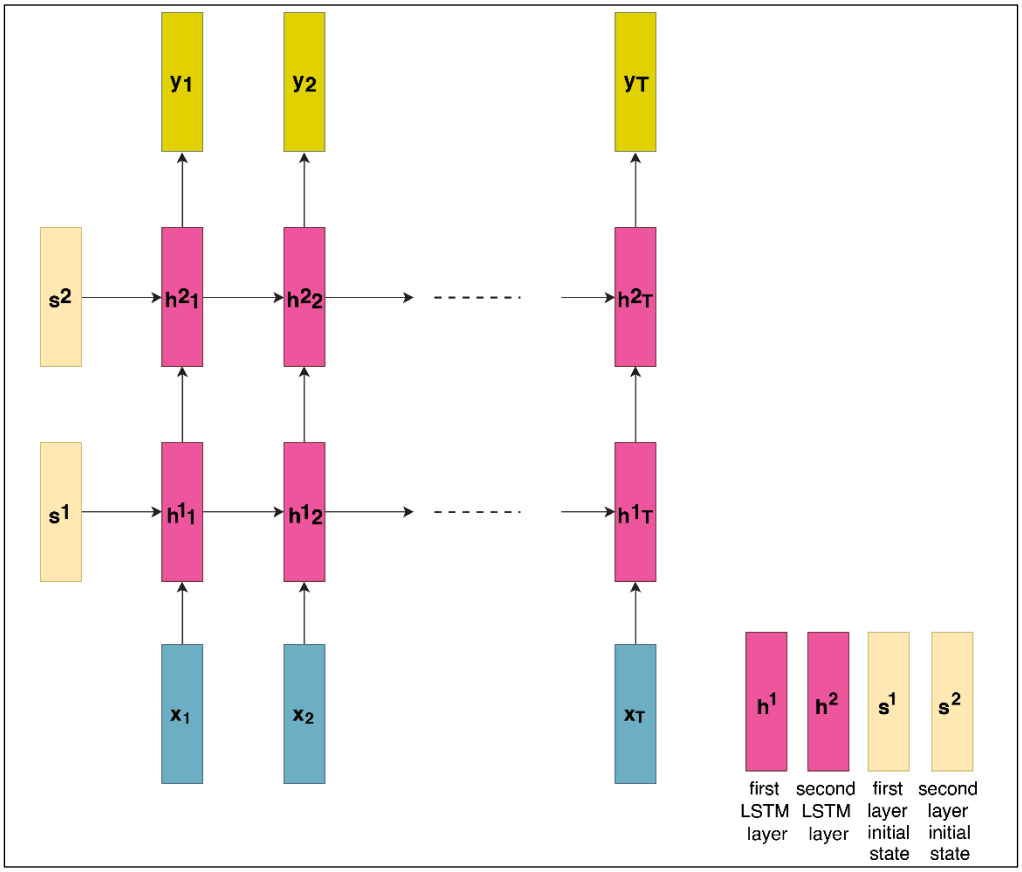

RNN (Time-Unfolded)
RNN (Time-Folded)
y1
y2
yT
y
s
h1
h2
hT
h
x1
x2
xT
x

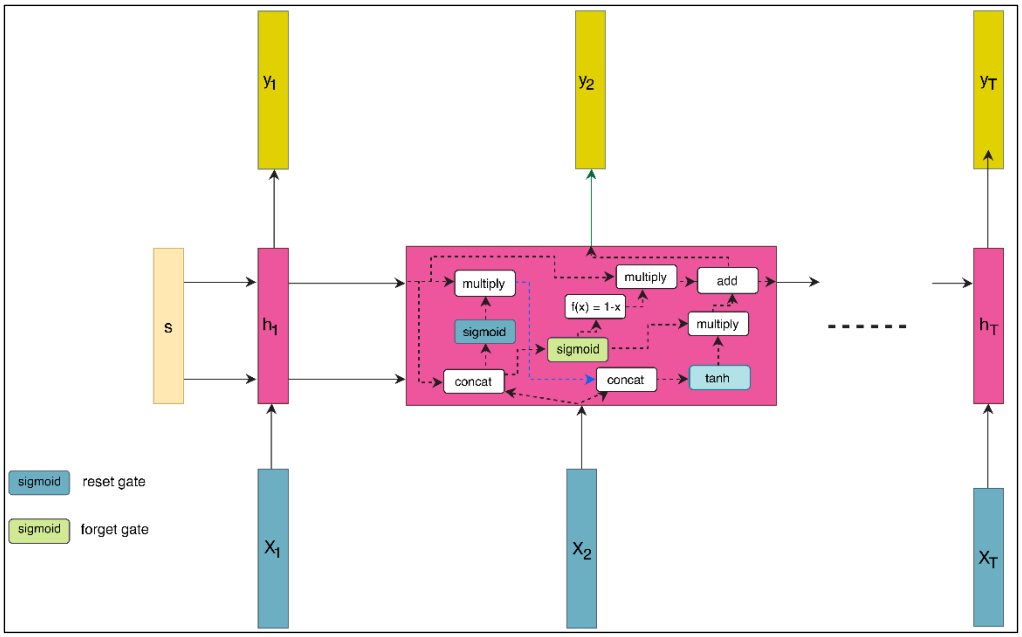

RNN
(1986)
BidirectionalRNN
(1997)
LSTM
(1997)
ExtendedLSTM
(2000)
StackedLSTM
(2013)
MultiDimensionalRNN
(2007)
BidirectionalLSTM
(2005)
GRU
(2014)
GridLSTM
(2015)
Gated Orthogonal
Recurrent Units
(2017)

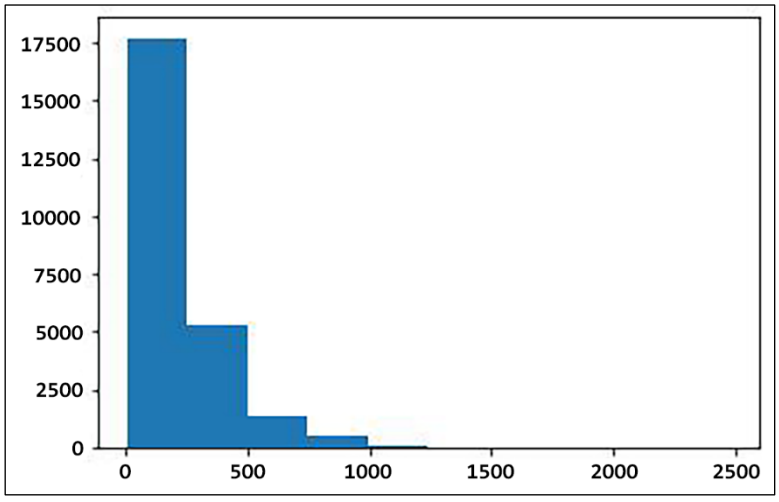

Bidirectional RNN
y1
y2
yT-1
yT-1
gT
gT-1
g2
g1
s
s
h1
h2
hT-1
hT-1
x1
x2
xT-1
xT-1
Forward RNN
Backward RNN

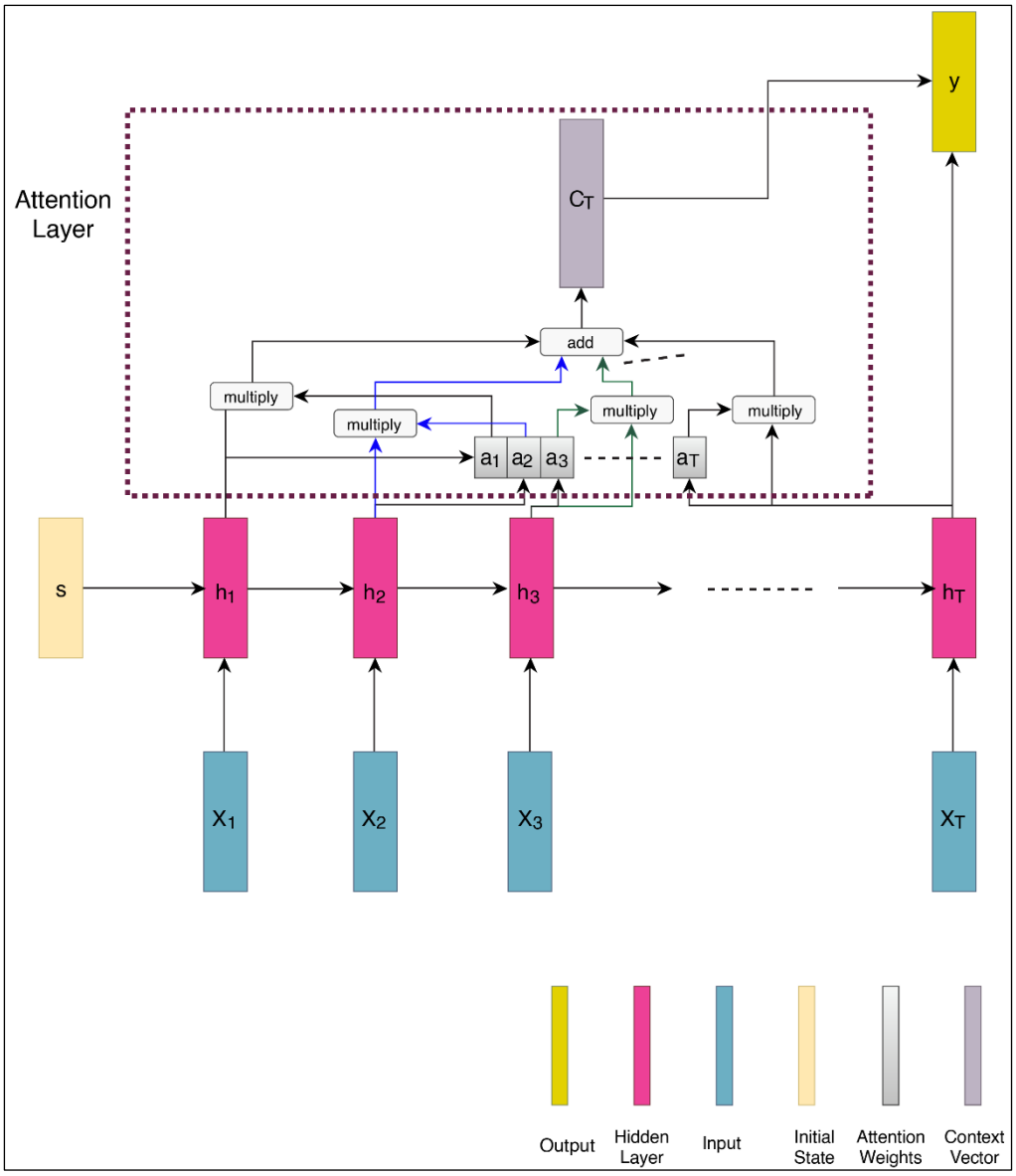

y1
y2
yT
s
h1
concat
tanh
hT
X1
X2
XT


y1
y2
yT
s
h1
internal state
external state
multiply
add
sigmoid
sigmoid
tanh
multiply
concat
sigmoid
hT
sigmoid forget gate
sigmoid output gate
sigmoid input gate
X1
X2
XT

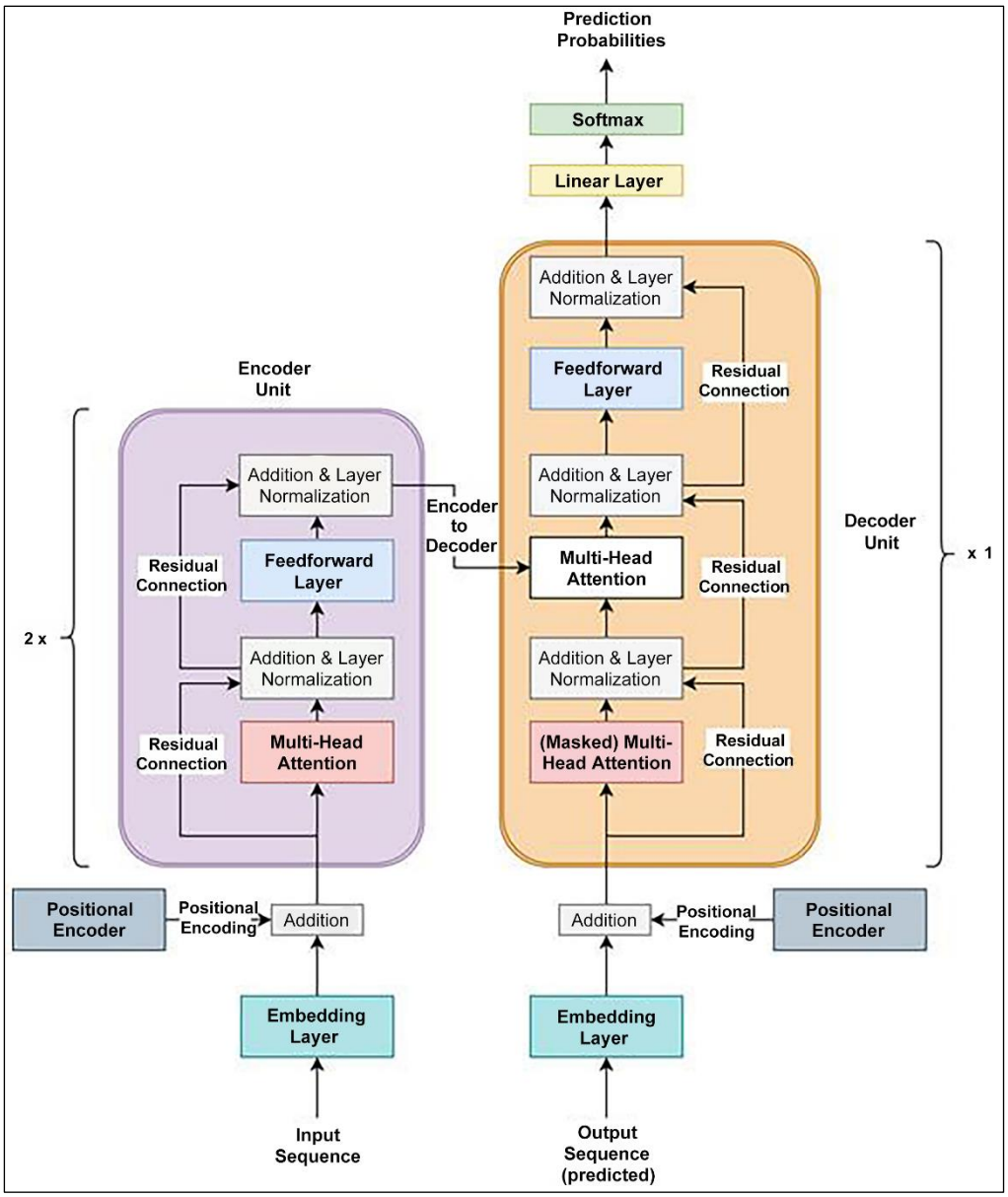

y1
y2
yT
s2
h2 1
h2 2
h2T
s1
h1 1
h1 2
h1T
x1
x2
xT
h1
h2
s1
s2
first LSTM layer
second LSTM layer
first layer initial state
second layer initial state

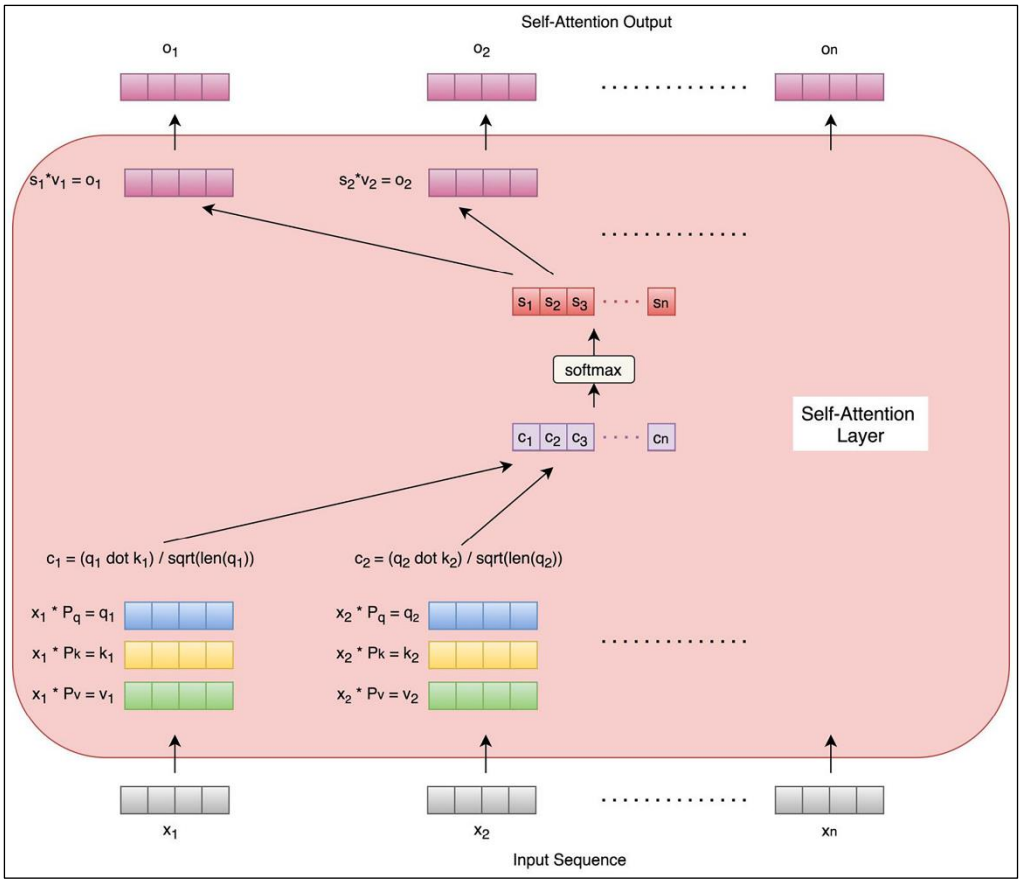

y1
y2
yT
s
h1
hT
multiply
multiply
add
f(x) = 1-x
sigmoid
multiply
sigmoid
concat
concat
tanh
sigmoid reset gate
sigmoid forget gate
X1
X2
XT

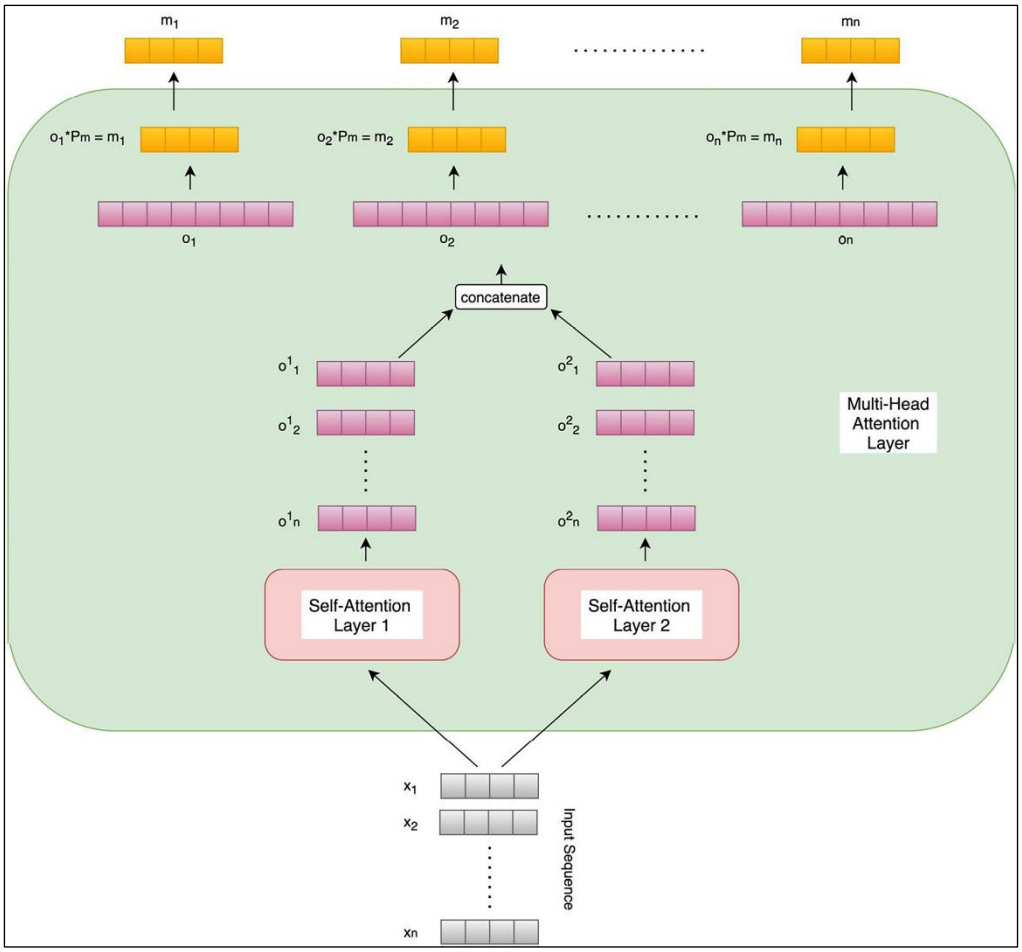

17500
15000
12500
10000
7500
5000
2500
0
0
500
1000
1500
2000
2500

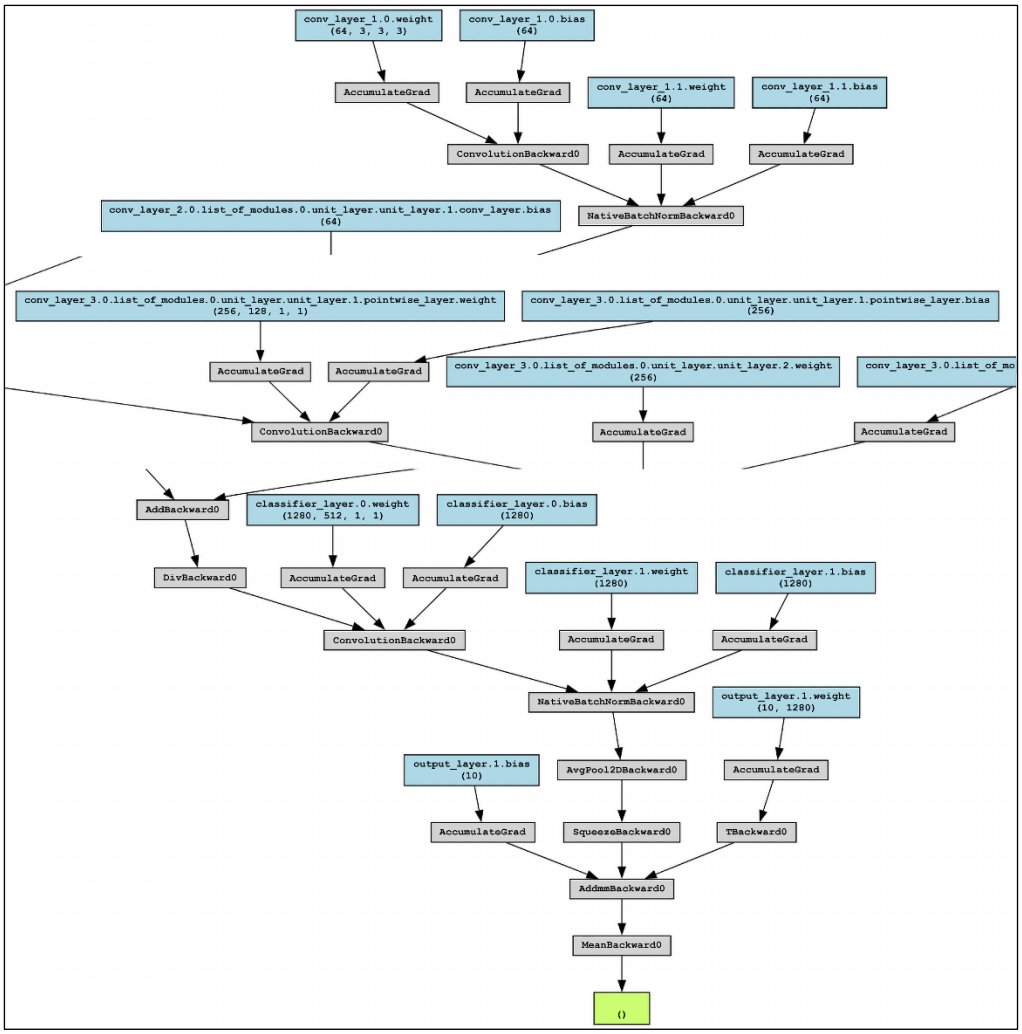

Attention Layer
y
C_T
add
multiply
multiply
multiply
multiply
a_1 a_2 a_3
a_T
s
h_1
h_2
h_3
h_T
X_1
X_2
X_3
X_T
Output
Hidden Layer
Input
Initial State
Attention Weights
Context Vector

Chapter 5: Advanced Hybrid Models

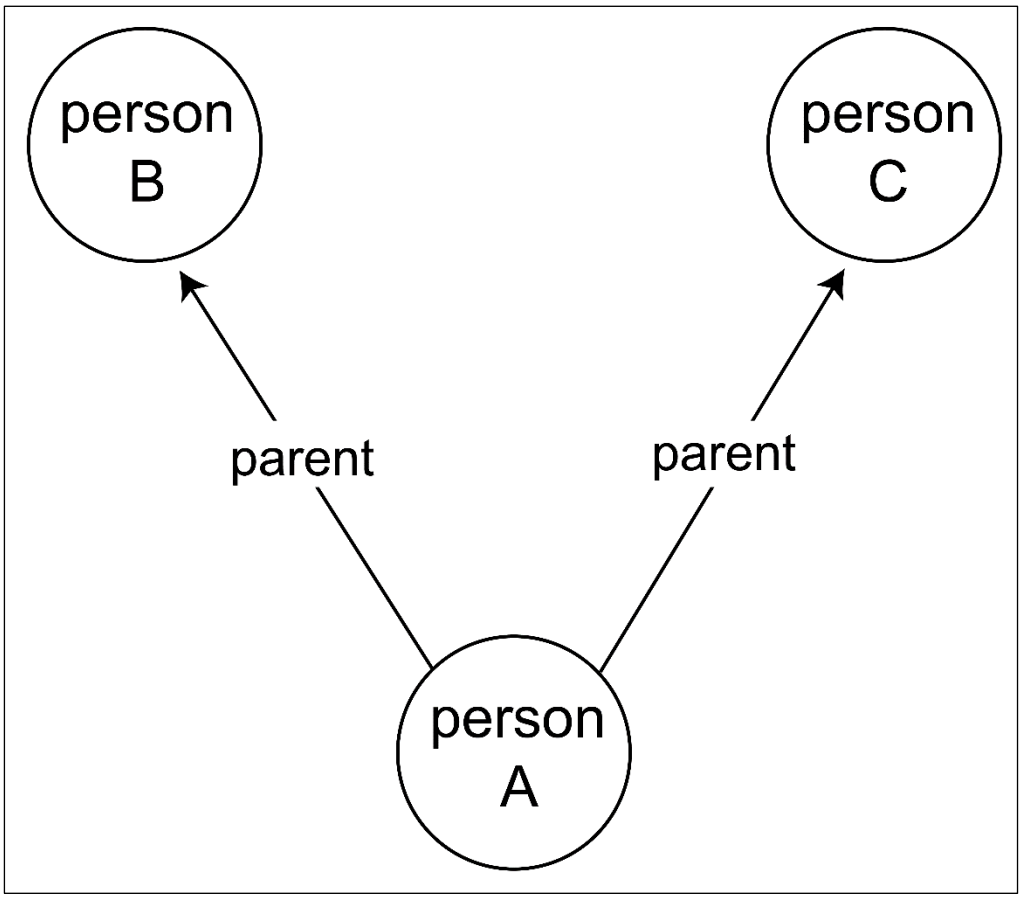


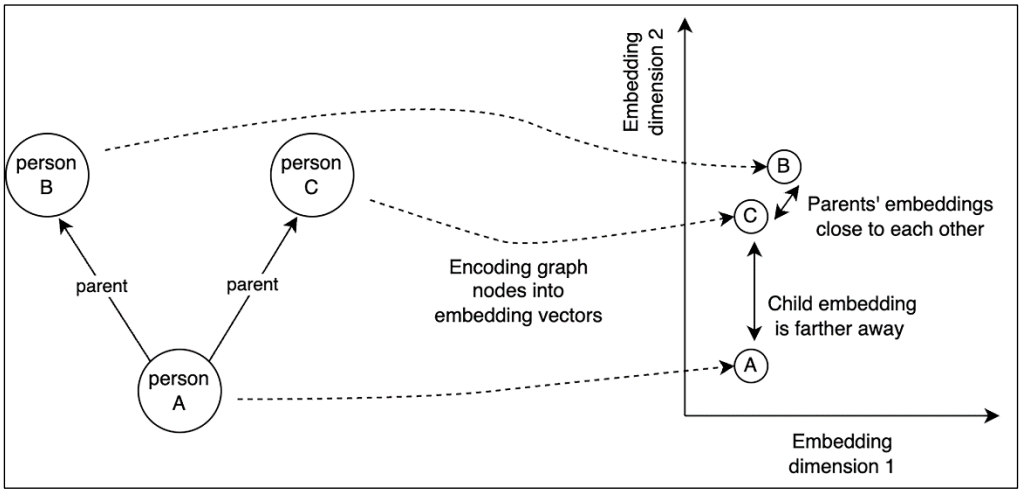

Self-Attention Output
o1
o2
on
s1*v1 = o1
s2*v2 = o2
s1 s2 s3 sn
softmax
c1 c2 c3 cn
Self-Attention Layer
c1 = (q1 dot k1) / sqrt(len(q1))
c2 = (q2 dot k2) / sqrt(len(q2))
x1 * Pq = q1
x1 * Pk = k1
x1 * Pv = v1
x2 * Pq = q2
x2 * Pk = k2
x2 * Pv = v2
x1
x2
xn
Input Sequence

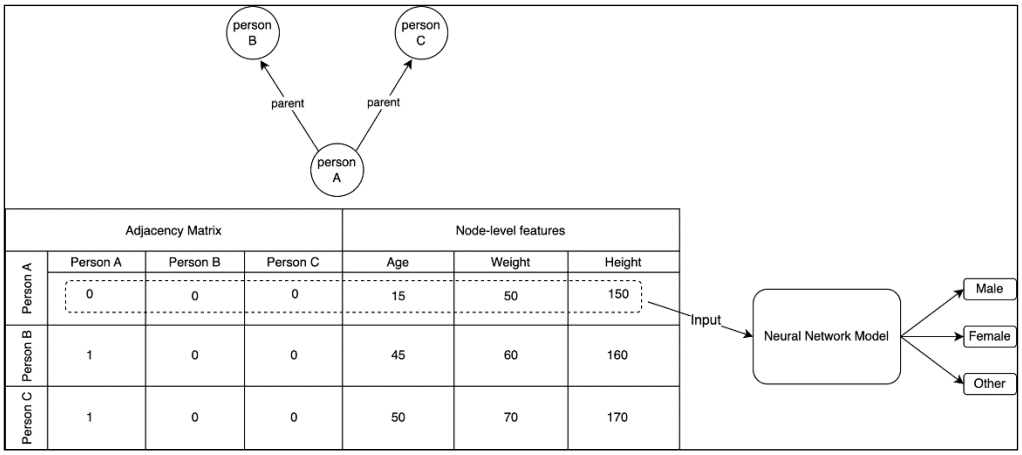

m1
m2
mn
o1*Pm = m1
o2*Pm = m2
on*Pm = mn
o1
o2
on
concatenate
o1 1
o1 2
o1 n
o2 1
o2 2
o2 n
Multi-Head Attention Layer
Self-Attention Layer 1
Self-Attention Layer 2
x1
x2
xn
Input Sequence

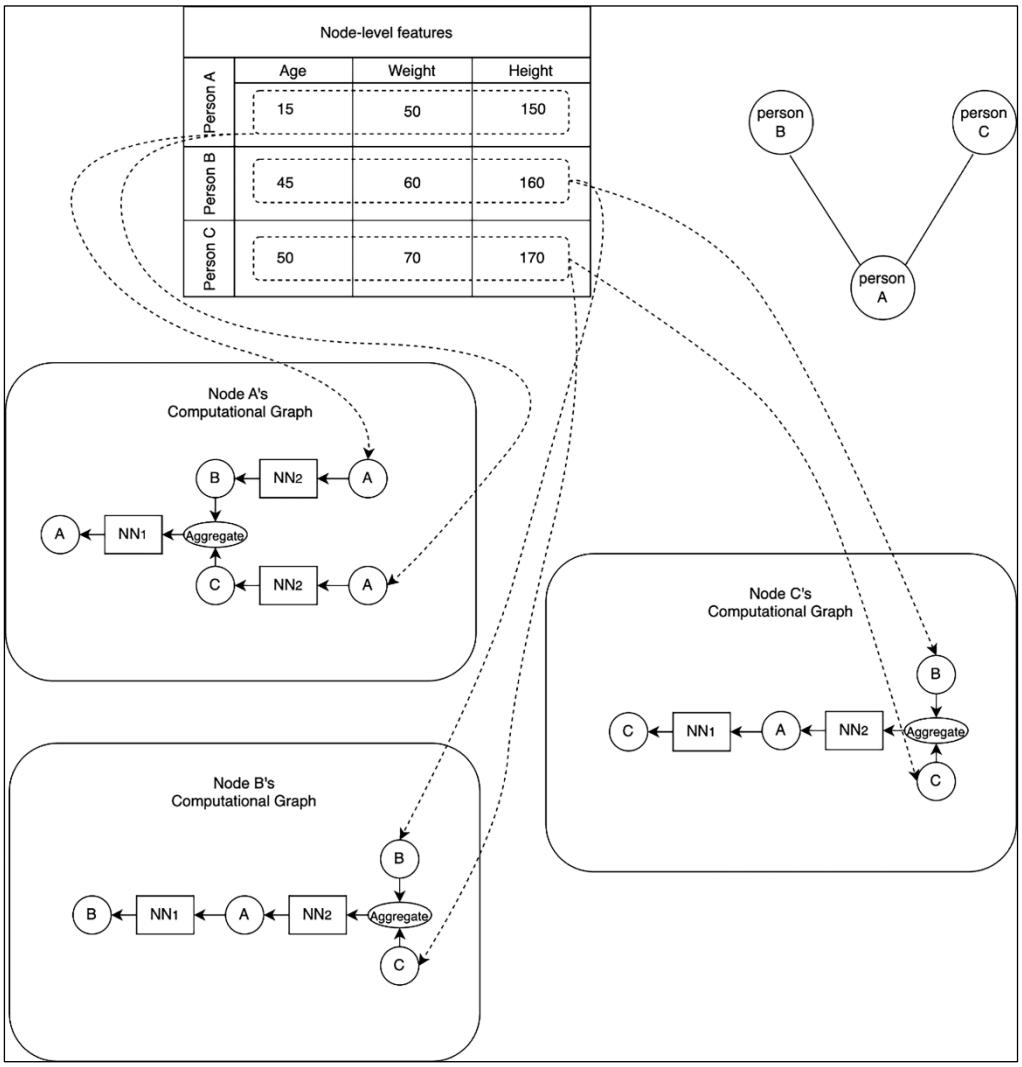

conv_layer_1.0.weight
(64, 3, 3, 3)
conv_layer_1.0.bias
(64)
AccumulateGrad
AccumulateGrad
conv_layer_1.1.weight
(64)
conv_layer_1.1.bias
(64)
ConvolutionBackward0
AccumulateGrad
AccumulateGrad
conv_layer_2.0.list_of_modules.0.unit_layer.unit_layer.1.conv_layer.bias
(64)
NativeBatchNormBackward0
conv_layer_3.0.list_of_modules.0.unit_layer.unit_layer.1.pointwise_layer.weight
(256, 128, 1, 1)
conv_layer_3.0.list_of_modules.0.unit_layer.unit_layer.1.pointwise_layer.bias
(256)
AccumulateGrad
AccumulateGrad
conv_layer_3.0.list_of_modules.0.unit_layer.unit_layer.2.weight
(256)
conv_layer_3.0.list_of_mo
ConvolutionBackward0
AccumulateGrad
AccumulateGrad
AddBackward0
classifier_layer.0.weight
(1280, 512, 1, 1)
classifier_layer.0.bias
(1280)
DivBackward0
AccumulateGrad
AccumulateGrad
classifier_layer.1.weight
(1280)
classifier_layer.1.bias
(1280)
ConvolutionBackward0
AccumulateGrad
AccumulateGrad
NativeBatchNormBackward0
output_layer.1.weight
(10, 1280)
output_layer.1.bias
(10)
AvgPool2DBackward0
AccumulateGrad
AccumulateGrad
SqueezeBackward0
TBackward0
AddmmBackward0
MeanBackward0
()

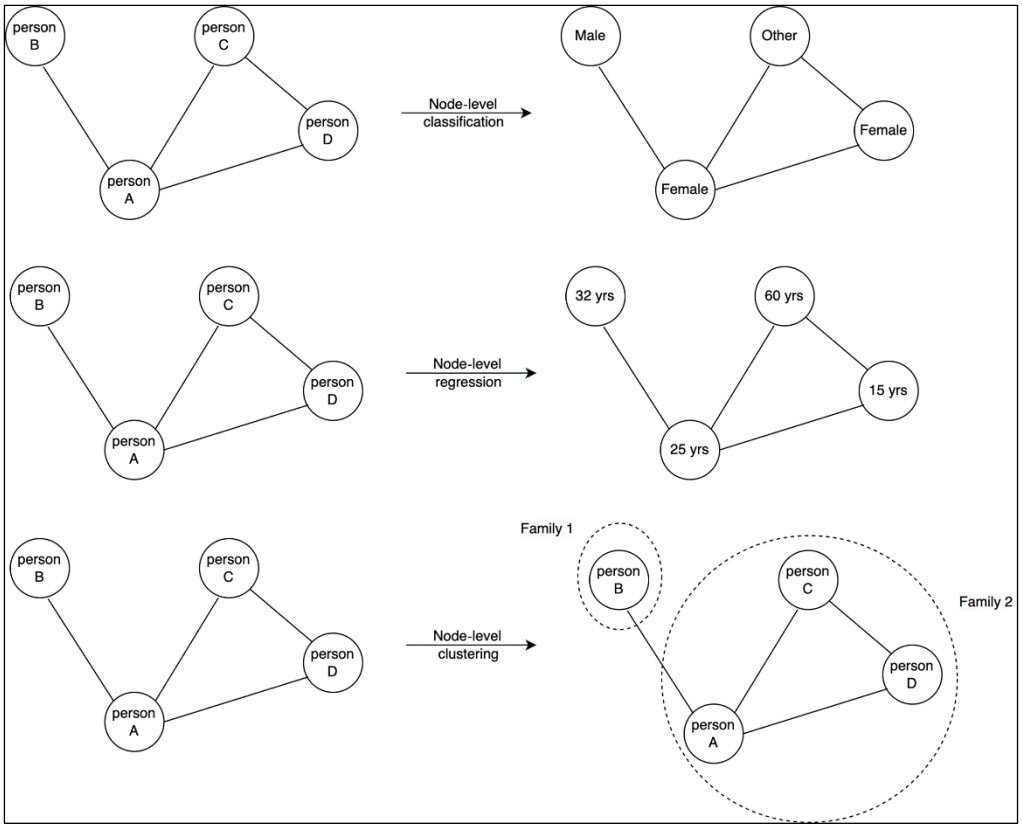
Chapter 6: Graph Neural Networks

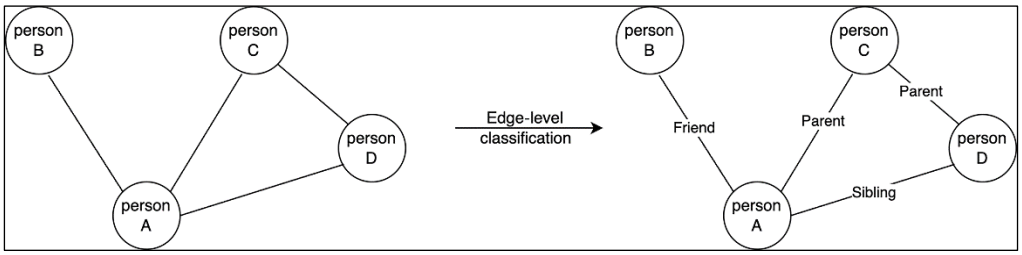


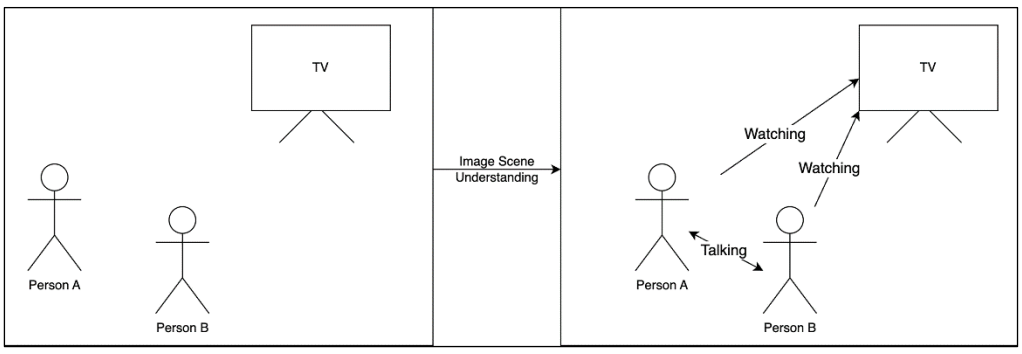


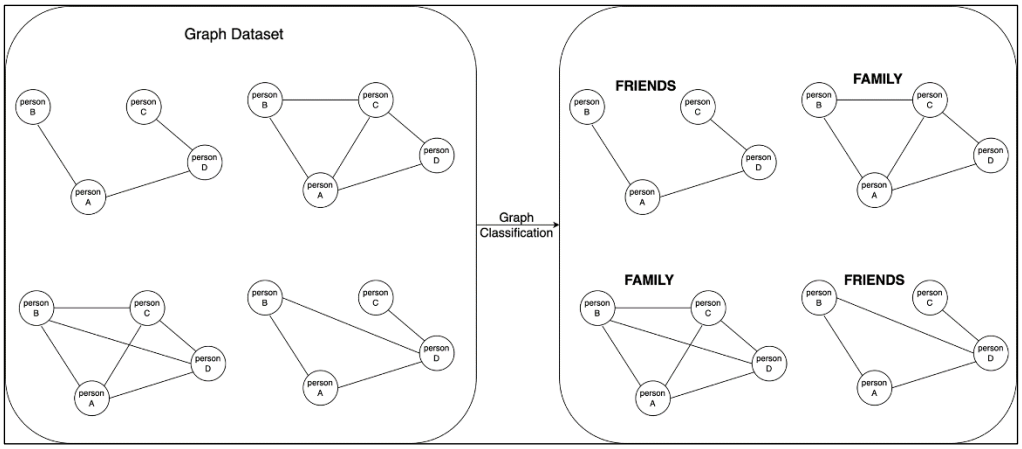

person B
person C
parent
parent
person A
Adjacency Matrix
Node-level features
Person A
Person B
Person C
Age
Weight
Height
0 0 0 15 50 150
1 0 0 45 60 160
1 0 0 50 70 170
Input
Neural Network Model
Male
Female
Other

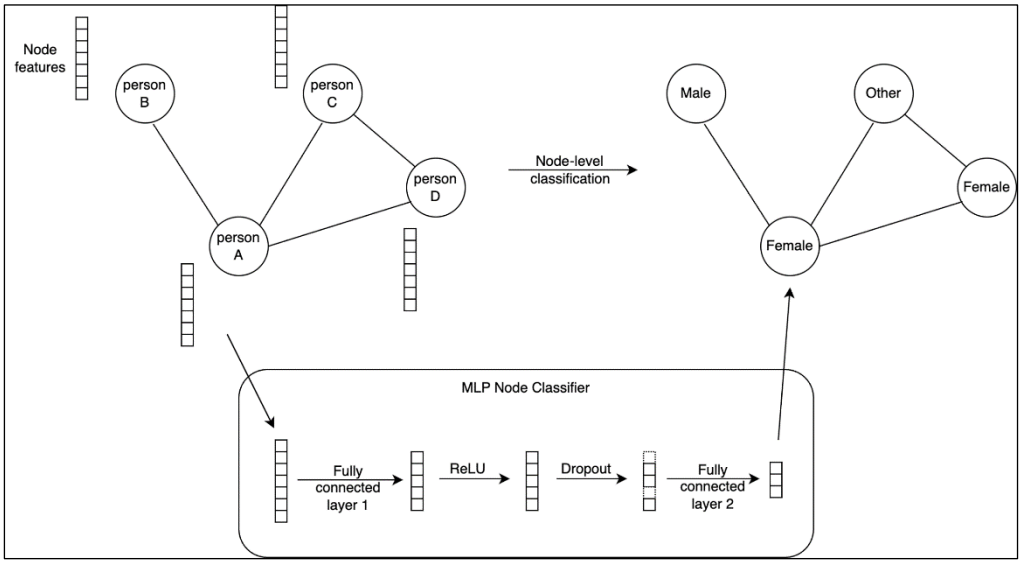

Node-level features
Person A
Person B
Person C
Age
Weight
Height
15 50 150
45 60 160
50 70 170
person B
person C
person A
Node A's Computational Graph
B
NN2
A
A
NN1
Aggregate
C
NN2
A
Node C's Computational Graph
B
C
NN1
A
NN2
Aggregate
C
Node B's Computational Graph
B
B
NN1
A
NN2
Aggregate
C

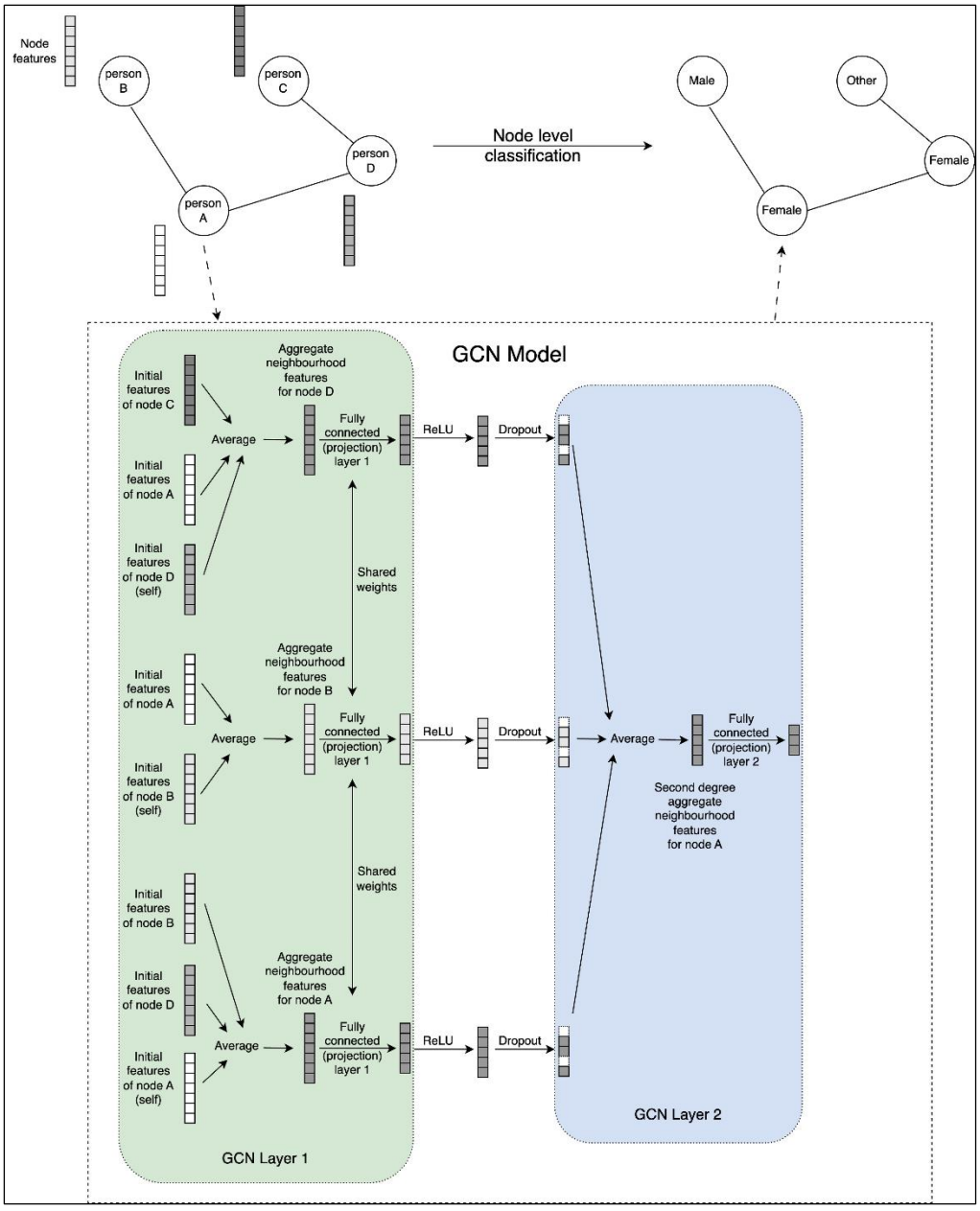

person B
person C
person D
person A
Node-level classification
Male
Other
Female
Female
person B
person C
person D
person A
Node-level regression
32 yrs
60 yrs
15 yrs
25 yrs
person B
person C
person D
person A
Node-level clustering
Family 1
person B
person C
person D
person A
Family 2

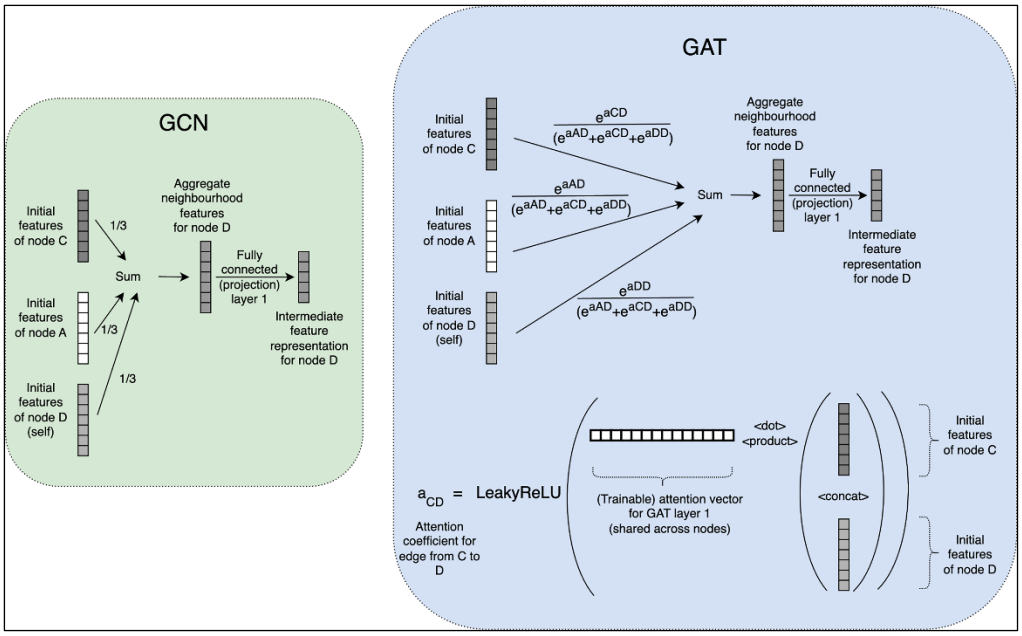

person B
person C
person D
person A
Edge-level classification
person B
person C
person D
person A
Friend
Parent
Parent
Sibling

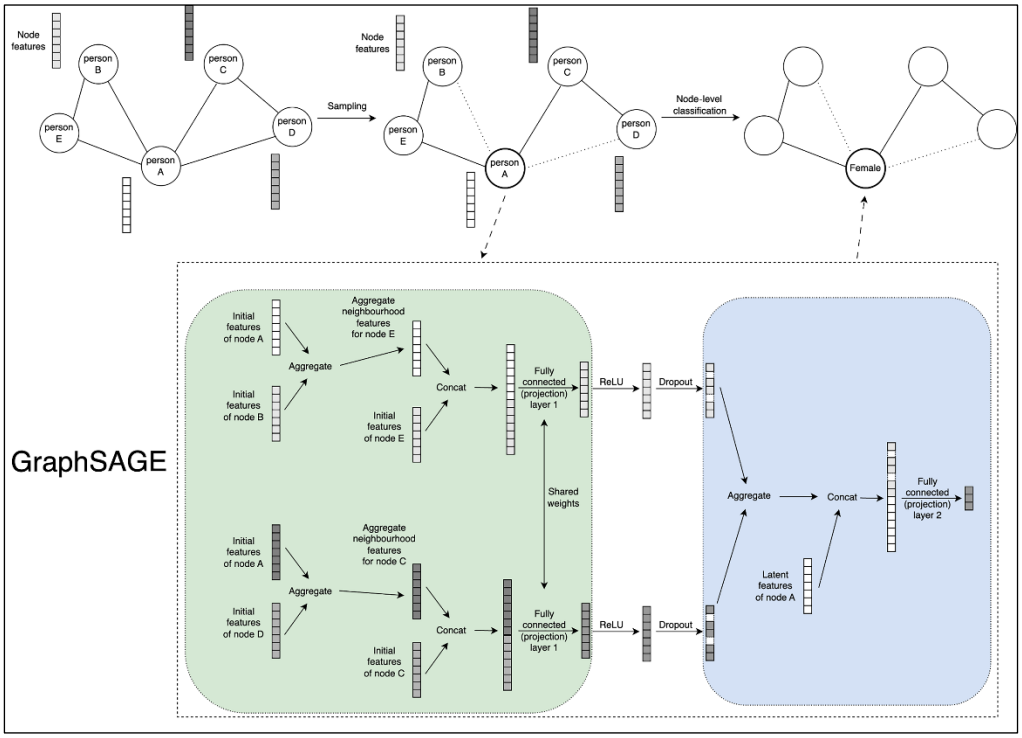

TV
Person A
Person B
Image Scene Understanding
TV
Watching
Watching
Talking
Person A
Person B

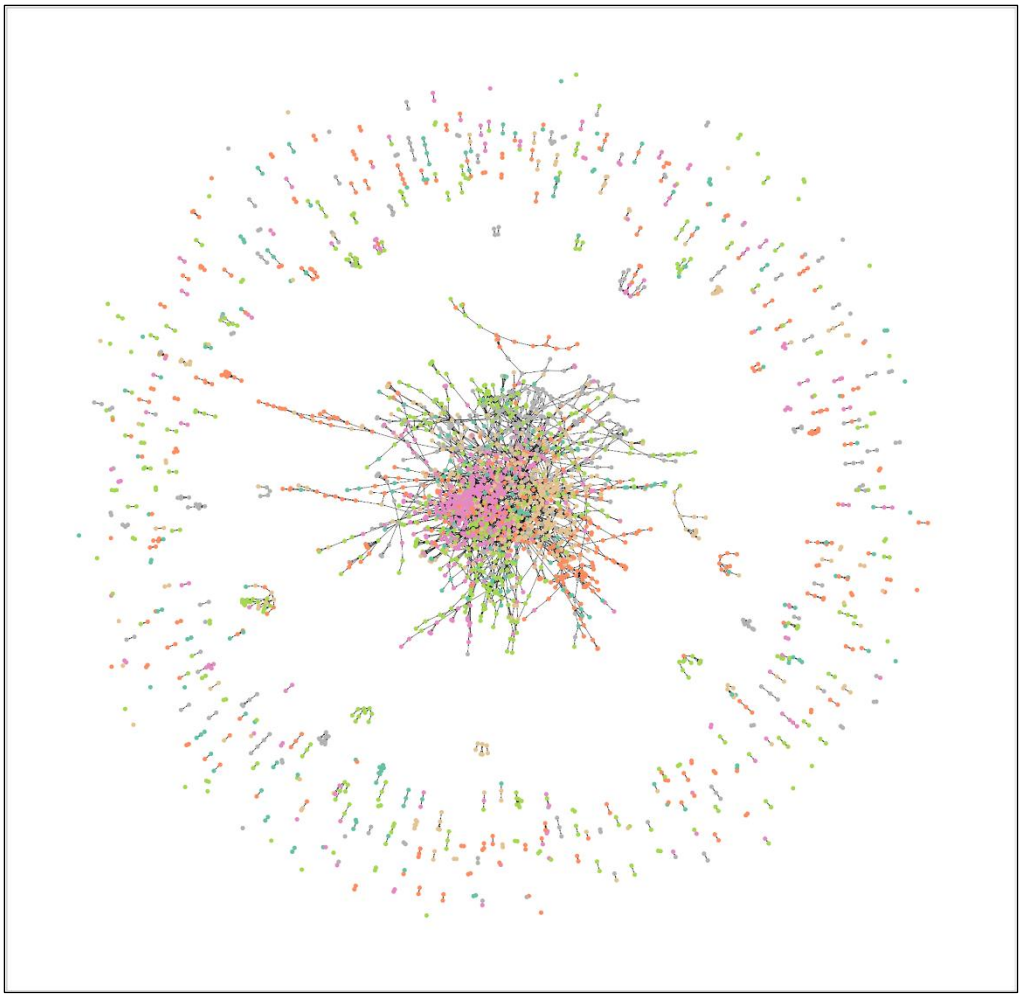

Graph Dataset
person B
person C
person D
person A
Graph Classification
FRIENDS
FAMILY
FAMILY
FRIENDS

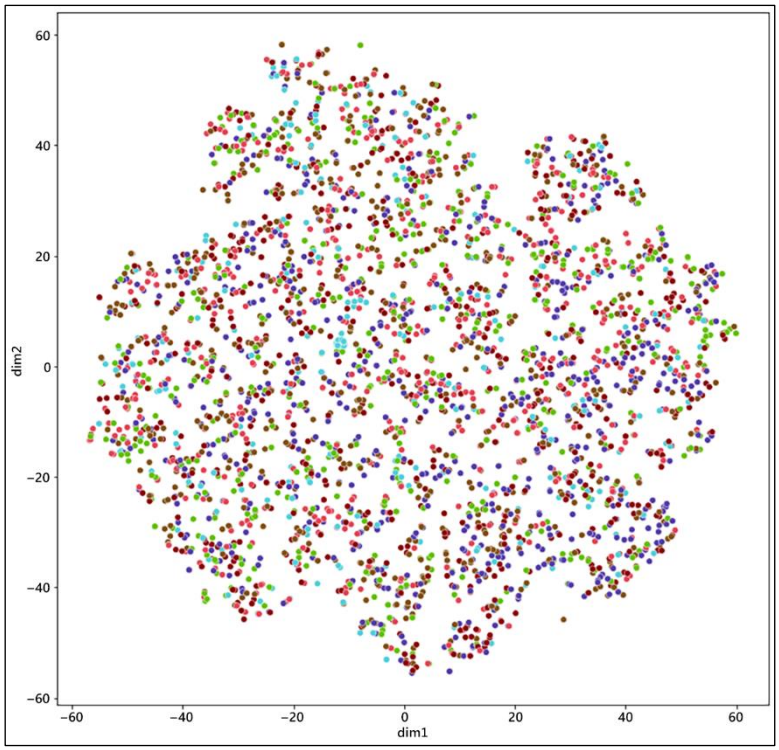

Node features
person B
person C
person D
person A
Node-level classification
Male
Other
Female
Female
MLP Node Classifier
Fully connected layer 1
ReLU
Dropout
Fully connected layer 2

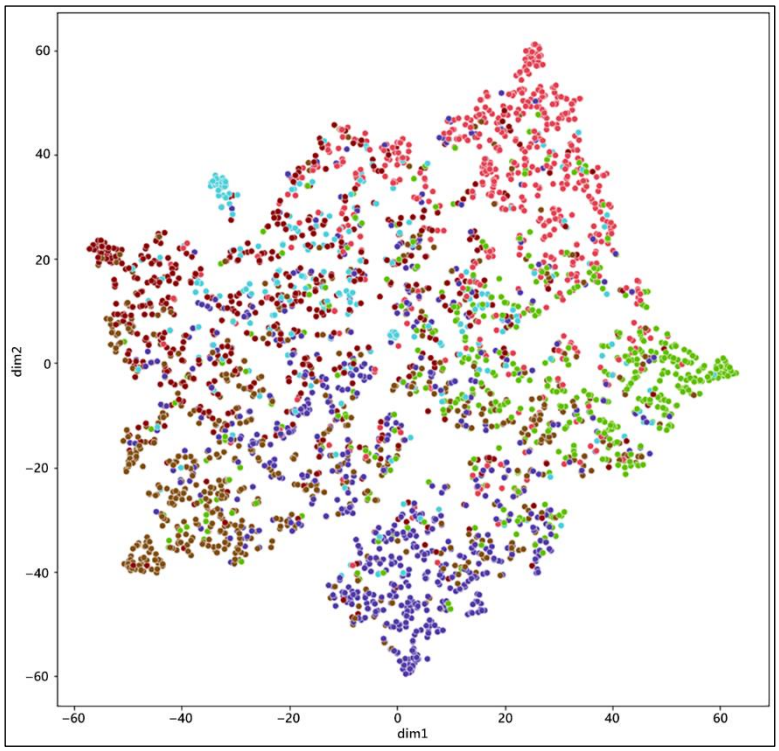

Node features
person B
person C
person D
person A
Node level classification
Male
Other
Female
Female
GCN Model
Initial features of node C
Initial features of node A
Initial features of node D (self)
Aggregate neighbourhood features for node D
Average
Fully connected (projection) layer 1
ReLU
Dropout
Shared weights
Initial features of node A
Initial features of node B (self)
Aggregate neighbourhood features for node B
Initial features of node B
Initial features of node D
Initial features of node A (self)
Aggregate neighbourhood features for node A
GCN Layer 1
Second degree aggregate neighbourhood features for node A
GCN Layer 2

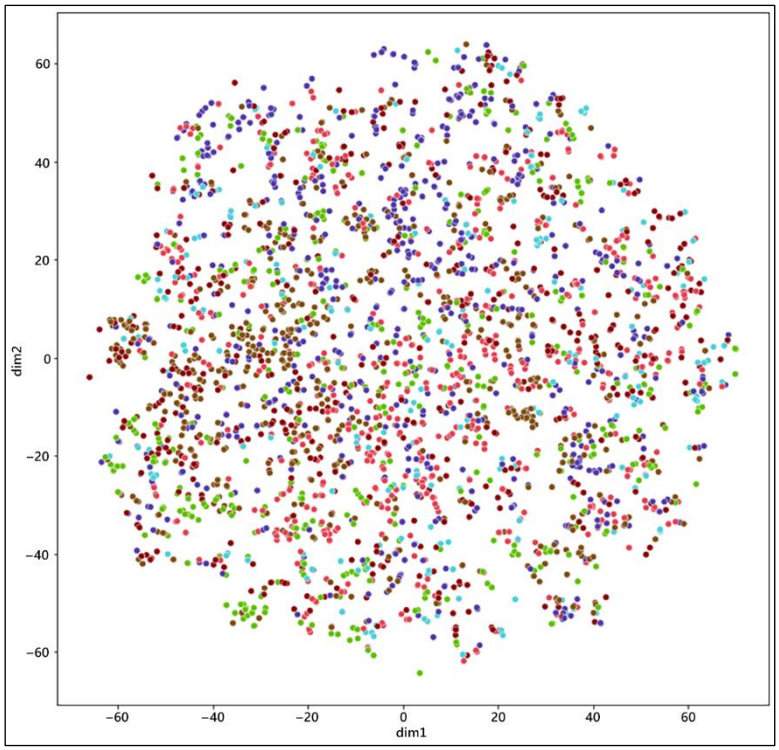

GCN
Initial features of node C
Initial features of node A
Initial features of node D (self)
1/3
1/3
1/3
Sum
Aggregate neighbourhood features for node D
Fully connected (projection) layer 1
Intermediate feature representation for node D
GAT
Initial features of node C
Initial features of node A
Initial features of node D (self)
Sum
Aggregate neighbourhood features for node D
Fully connected (projection) layer 1
Intermediate feature representation for node D
a_CD = LeakyReLU
Attention coefficient for edge from C to D
(Trainable) attention vector for GAT layer 1 (shared across nodes)
<dot> <product>
<concat>
Initial features of node C
Initial features of node D

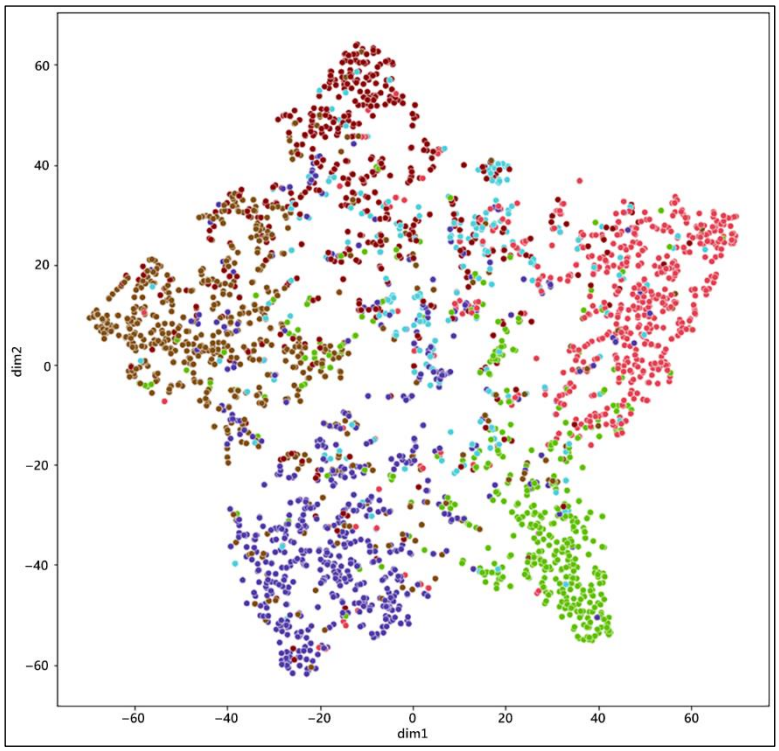

Node features
person B
person C
person D
person E
person A
Sampling
Node-level classification
Female
GraphSAGE
Initial features of node A
Initial features of node B
Aggregate
Aggregate neighbourhood features for node E
Concat
Initial features of node E
Fully connected (projection) layer 1
ReLU
Dropout
Shared weights
Initial features of node A
Initial features of node D
Aggregate
Aggregate neighbourhood features for node C
Concat
Initial features of node C
Fully connected (projection) layer 1
ReLU
Dropout
Aggregate
Concat
Fully connected (projection) layer 2
Latent features of node A

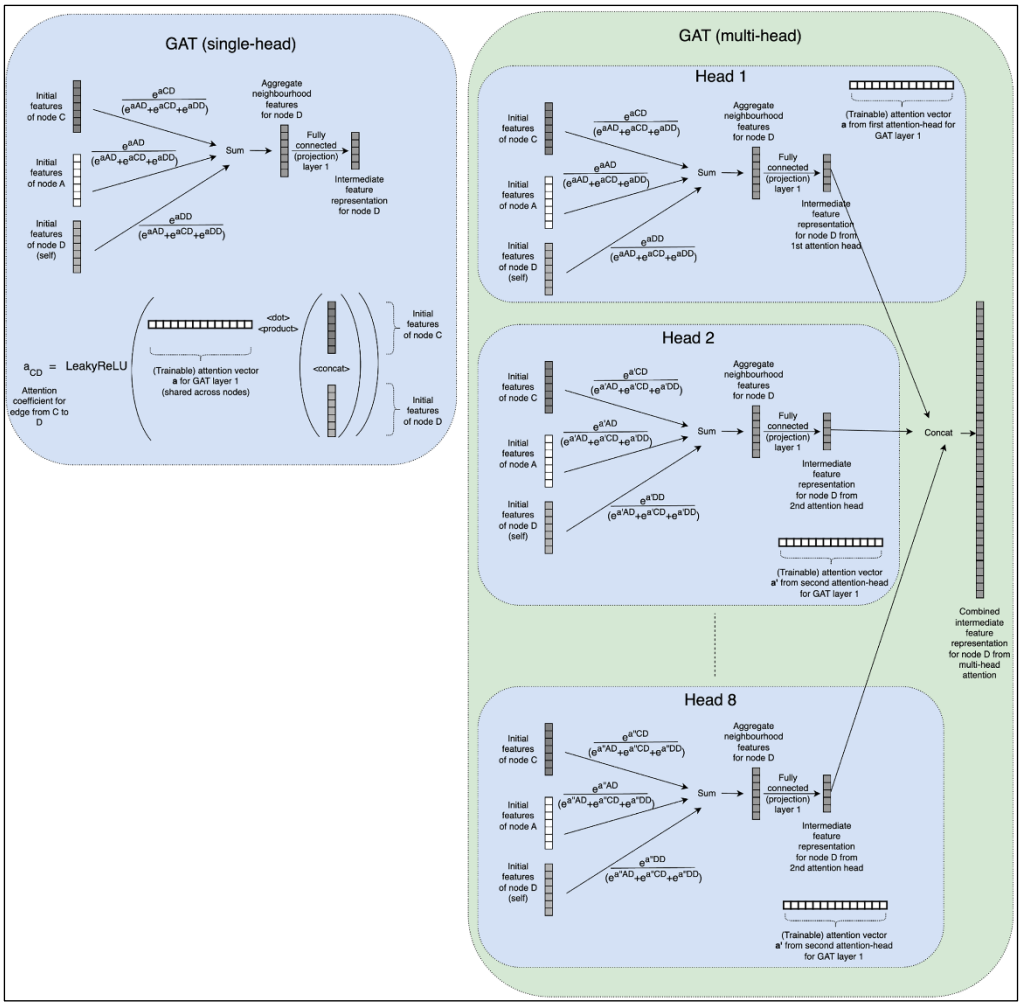


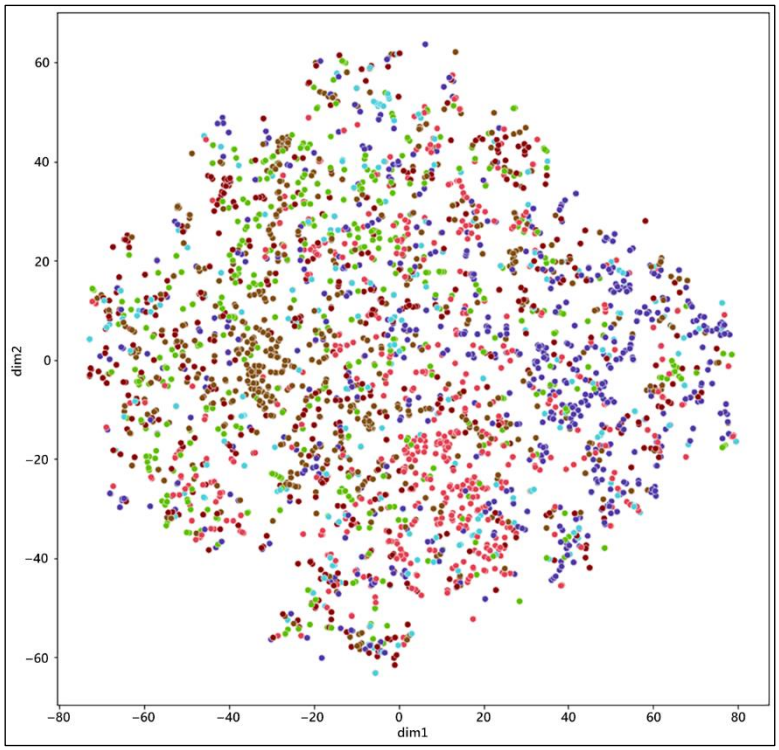

60
40
20
0
−20
−40
−60
dim2
−60
−40
−20
0
20
40
60
dim1

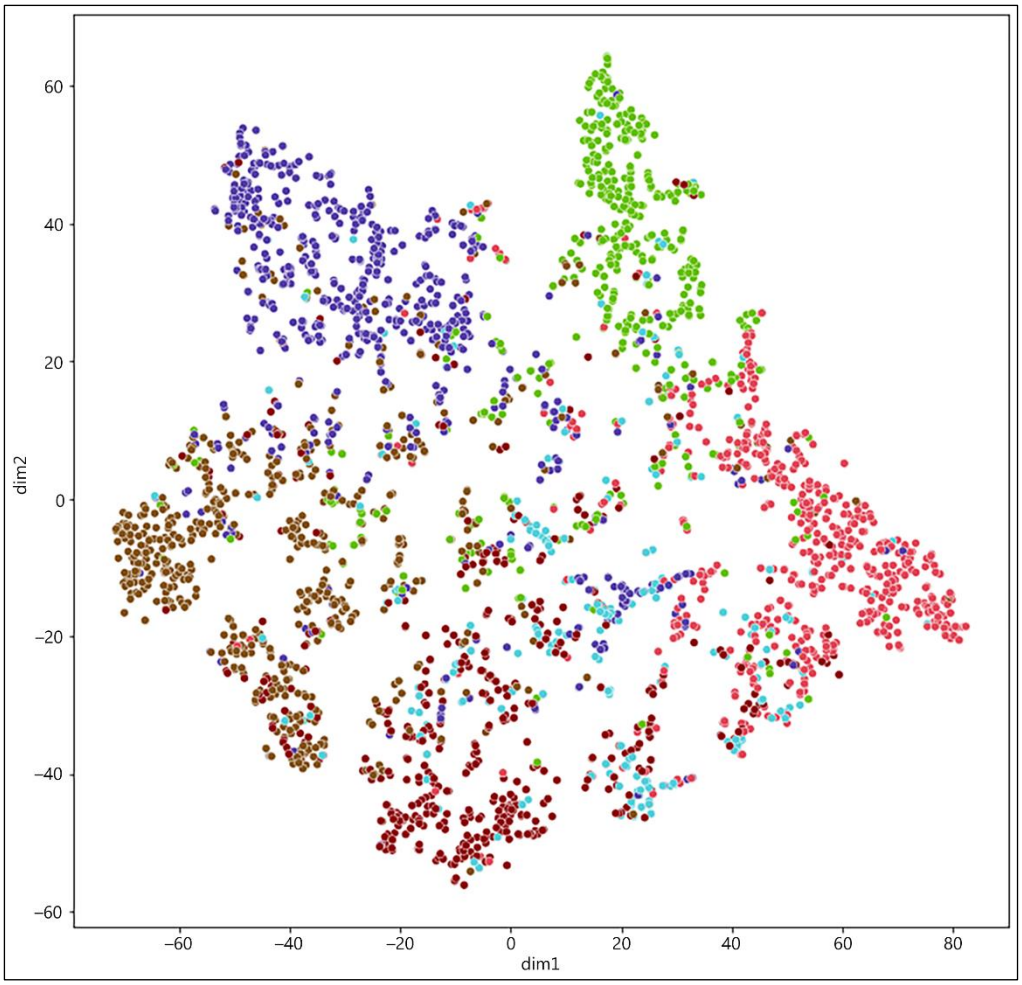

60
40
20
0
−20
−40
−60
dim2
−60
−40
−20
0
20
40
60
dim1


60
40
20
0
−20
−40
−60
dim2
−60
−40
−20
0
20
40
60
dim1

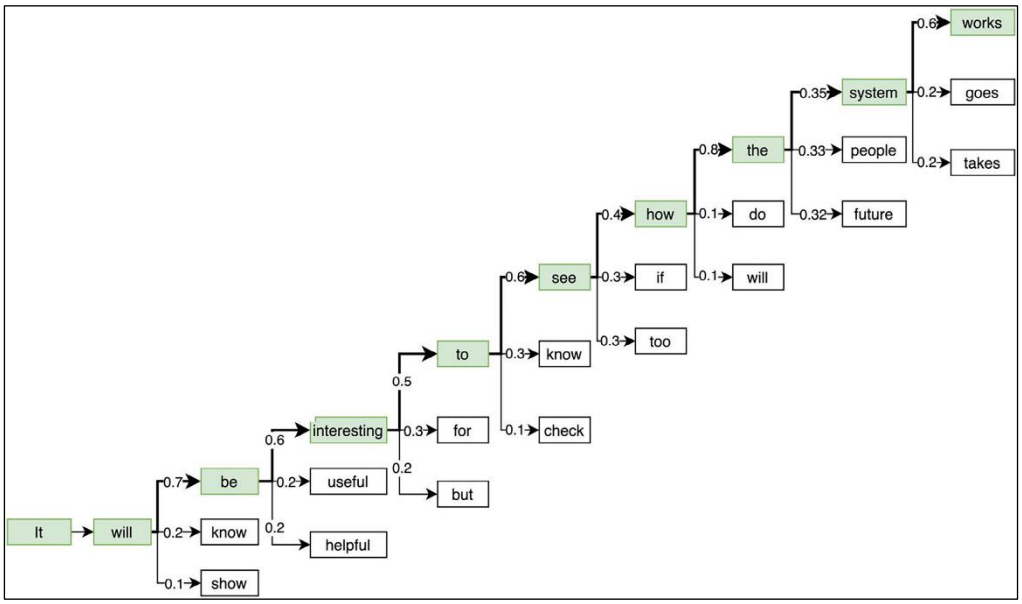

60
40
20
0
−20
−40
−60
dim2
−60
−40
−20
0
20
40
60
dim1

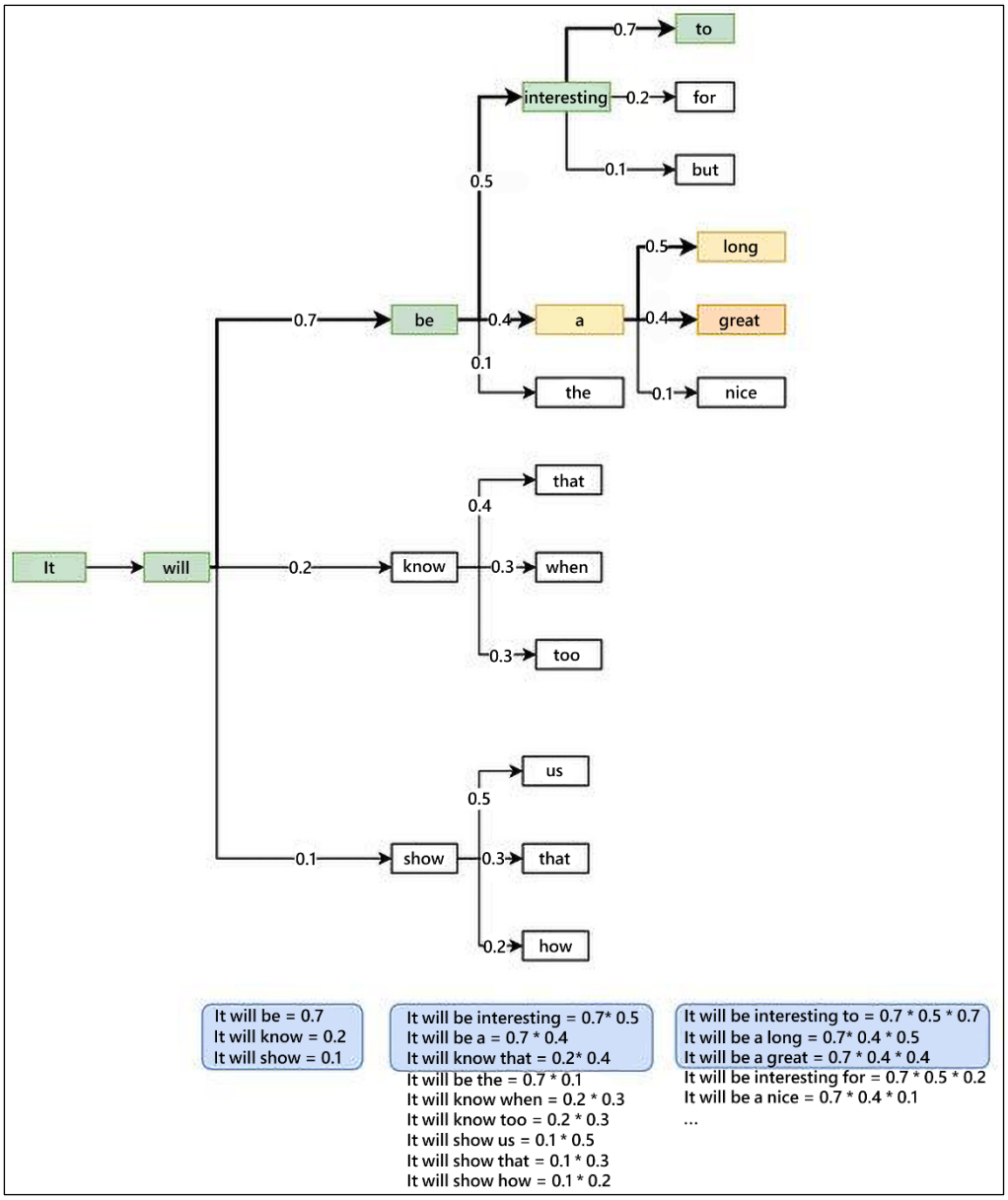

GAT (single-head)
Initial features of node C
Initial features of node A
Initial features of node D (self)
Sum
Aggregate neighbourhood features for node D
Fully connected (projection) layer 1
Intermediate feature representation for node D
a_CD = LeakyReLU
Attention coefficient for edge from C to D
(Trainable) attention vector a for GAT layer 1 (shared across nodes)
<dot> <product>
<concat>
Initial features of node C
Initial features of node D
GAT (multi-head)
Head 1
(Trainable) attention vector a from first attention-head for GAT layer 1
Intermediate feature representation for node D from 1st attention head
Head 2
(Trainable) attention vector a' from second attention-head for GAT layer 1
Intermediate feature representation for node D from 2nd attention head
Concat
Combined intermediate feature representation for node D from multi-head attention
Head 8
(Trainable) attention vector a'' from second attention-head for GAT layer 1
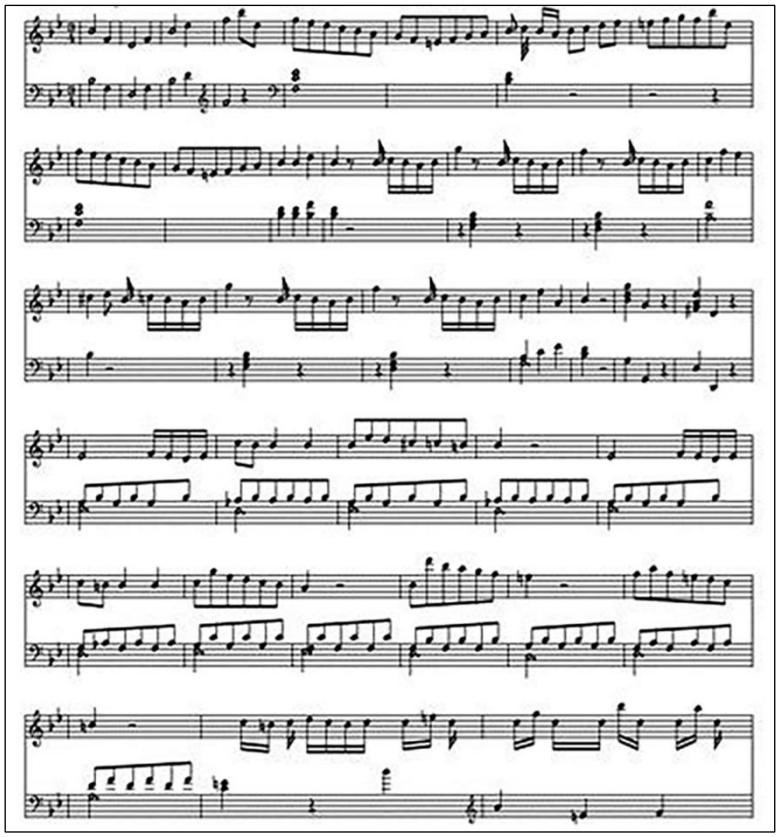

60
40
20
0
−20
−40
−60
dim2
−80
−60
−40
−20
0
20
40
60
80
dim1

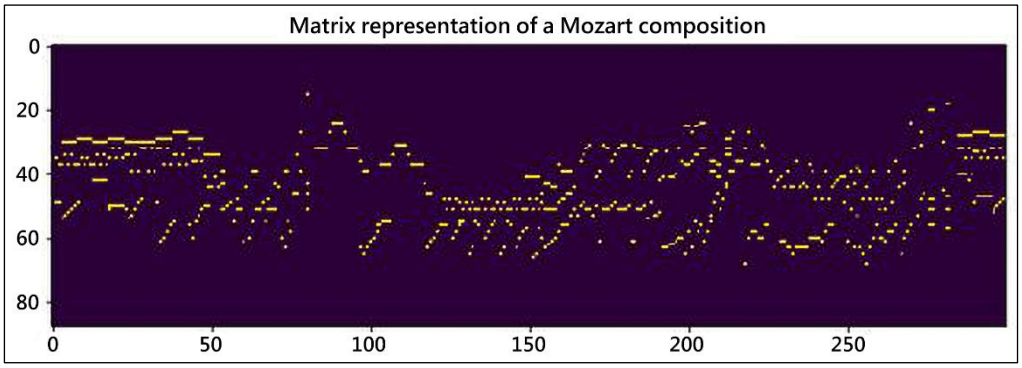

60
40
20
0
−20
−40
−60
dim2
−60
−40
−20
0
20
40
60
80
dim1

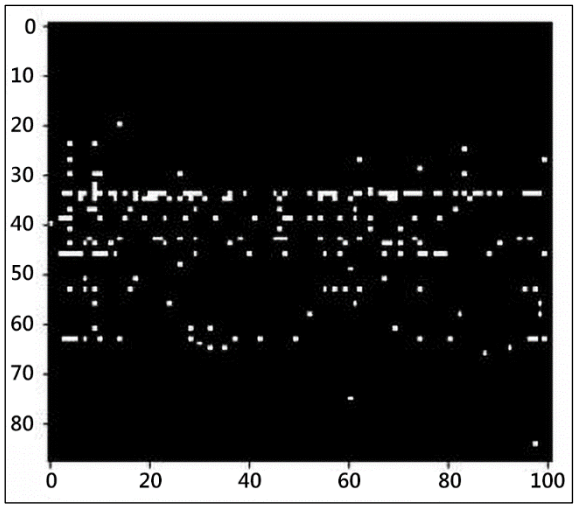
Chapter 7: Music and Text Generation with PyTorch
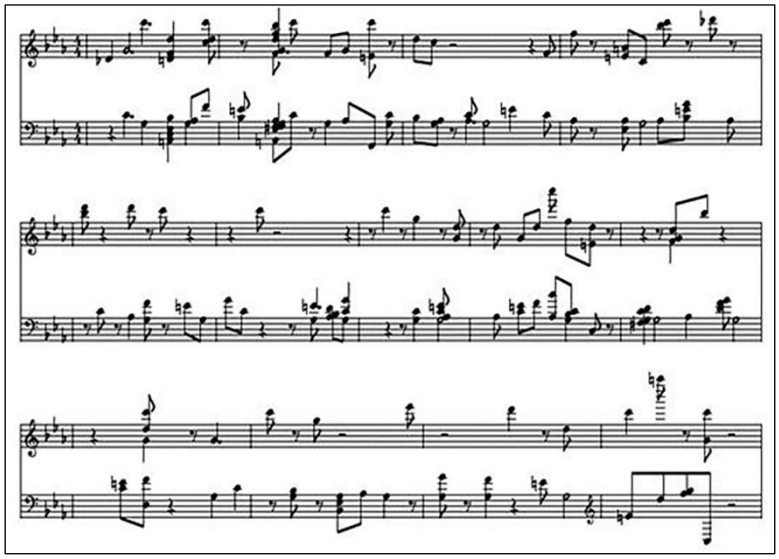



It
will
0.7
be
0.4
a
0.5
long
0.4
great
0.1
nice
0.1
the
0.5
interesting
0.7
to
0.2
for
0.1
but
0.2
know
0.4
that
0.3
when
0.3
too
0.1
show
0.5
us
0.3
that
0.2
how
It will be = 0.7
It will know = 0.2
It will show = 0.1
It will be interesting = 0.7* 0.5
It will be a = 0.7 * 0.4
It will know that = 0.2* 0.4
It will be the = 0.7 * 0.1
It will know when = 0.2 * 0.3
It will know too = 0.2 * 0.3
It will show us = 0.1 * 0.5
It will show that = 0.1 * 0.3
It will show how = 0.1 * 0.2
It will be interesting to = 0.7 * 0.5 * 0.7
It will be a long = 0.7* 0.4 * 0.5
It will be a great = 0.7 * 0.4 * 0.4
It will be interesting for = 0.7 * 0.5 * 0.2
It will be a nice = 0.7 * 0.4 * 0.1
...

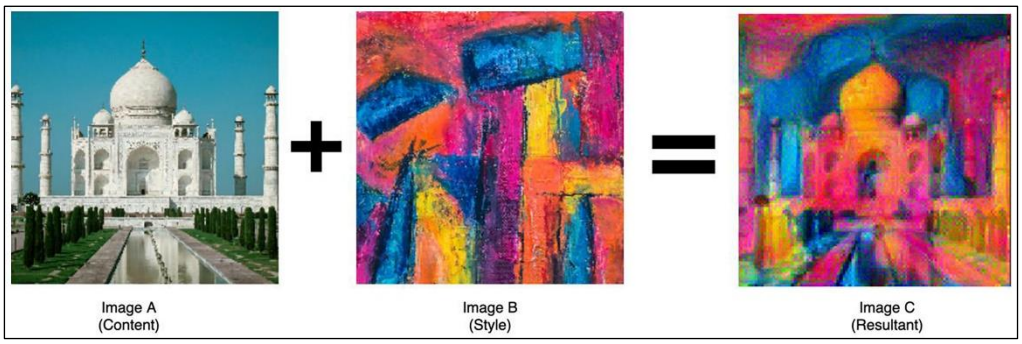


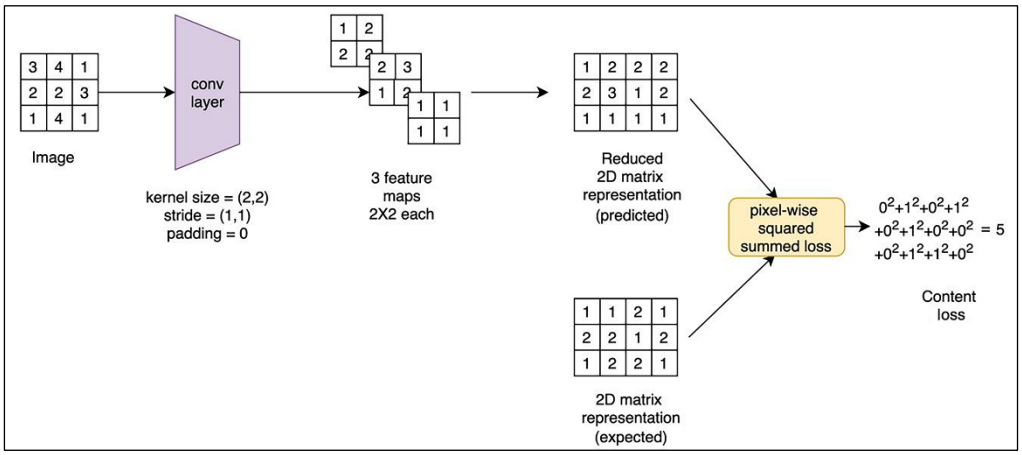

Matrix representation of a Mozart composition
0
20
40
60
80
0
50
100
150
200
250

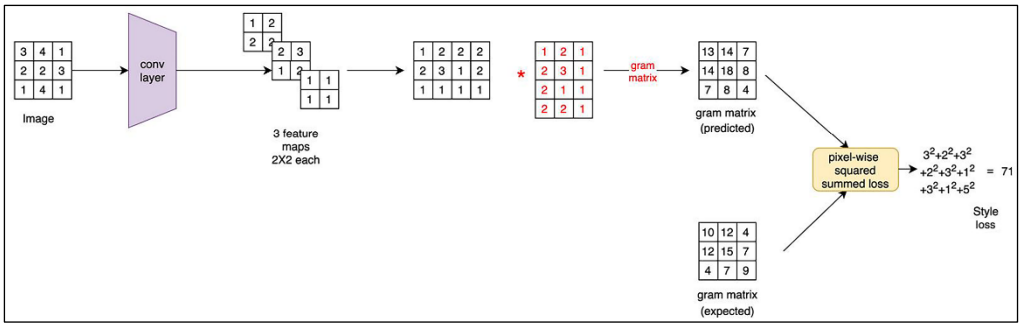

0
10
20
30
40
50
60
70
80
0
20
40
60
80
100

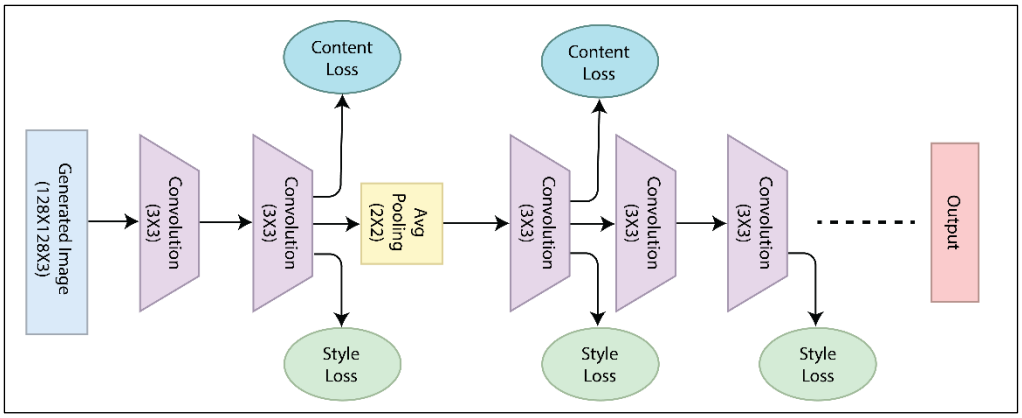


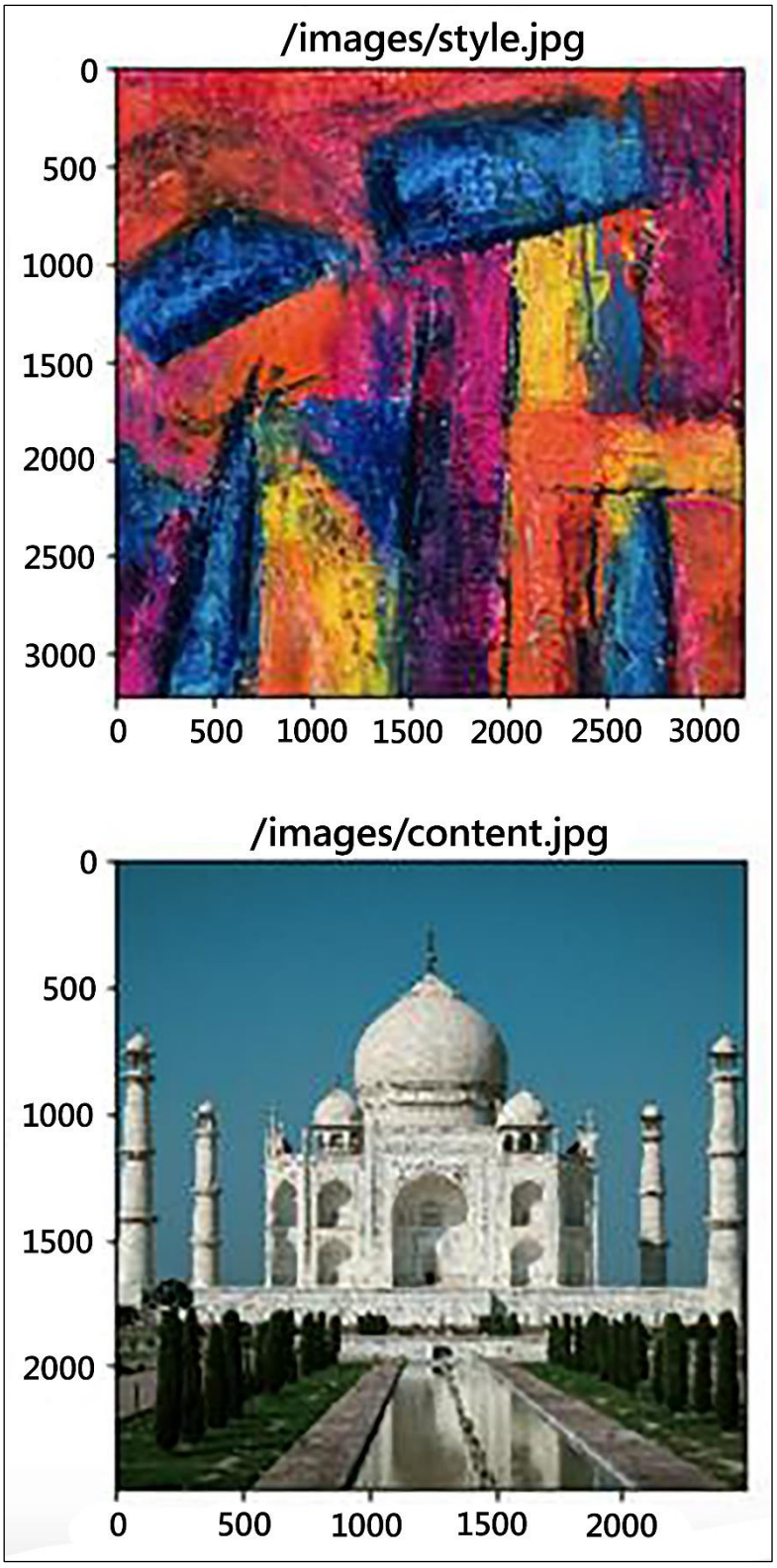
Chapter 8: Neural Style Transfer

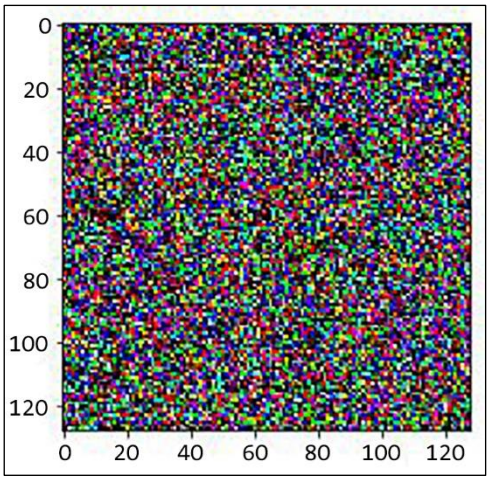


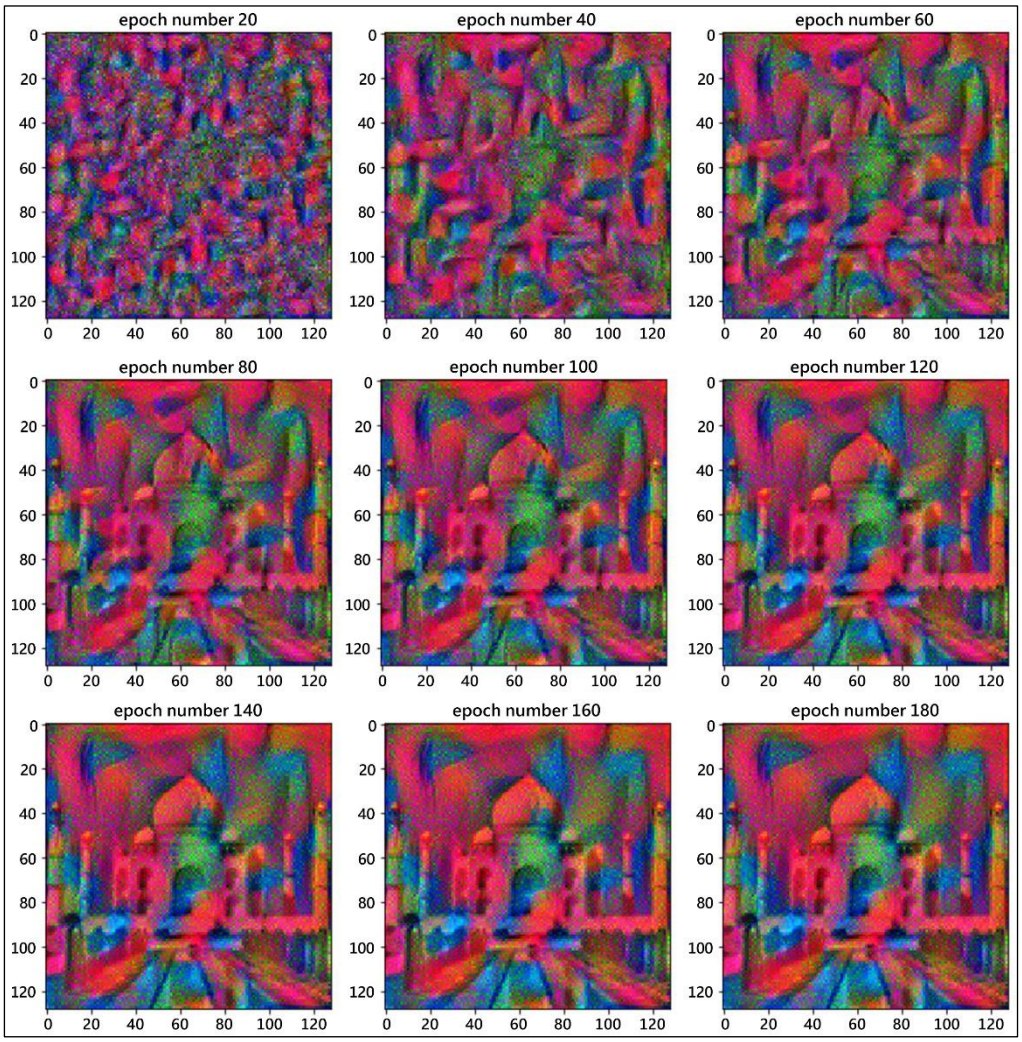


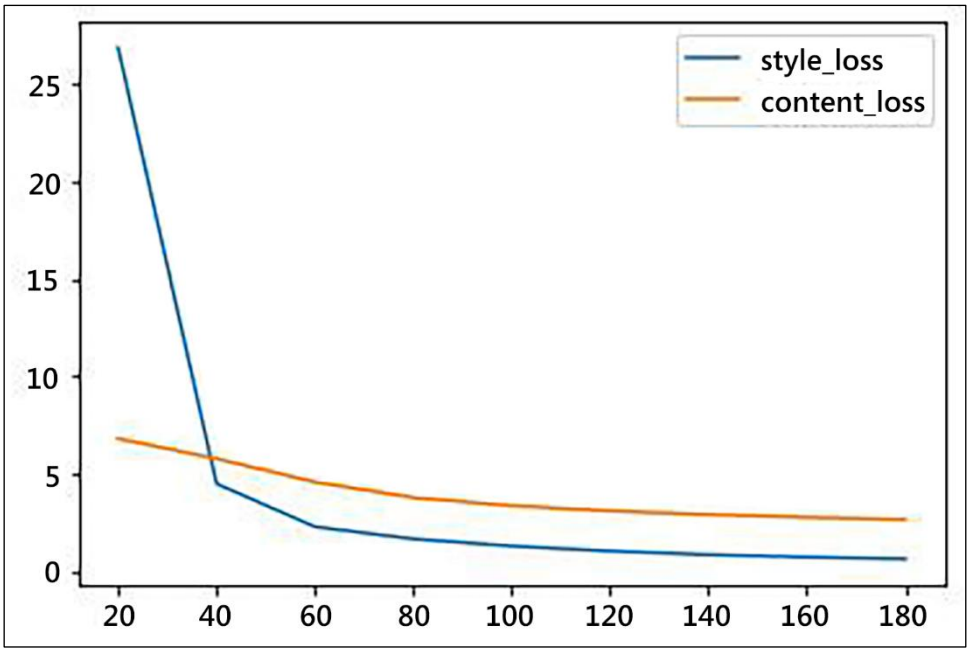


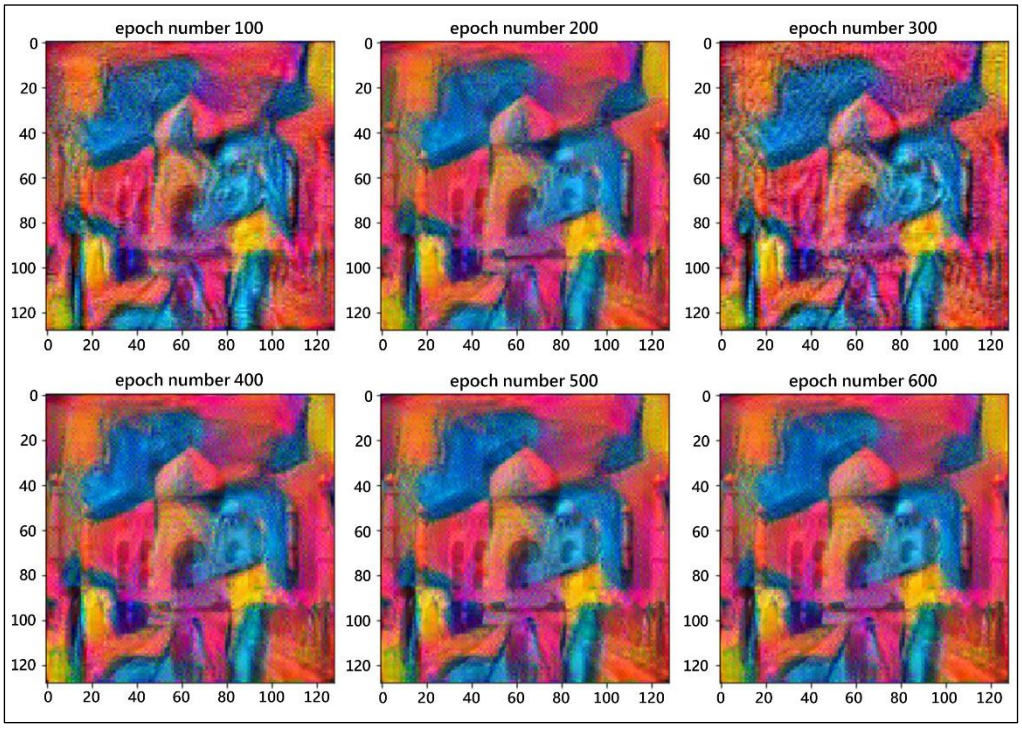

Generated Image (128X128X3)
Convolution (3X3)
Convolution (3X3)
Content Loss
Style Loss
Avg Pooling (2X2)
Convolution (3X3)
Convolution (3X3)
Content Loss
Style Loss
Convolution (3X3)
Style Loss
Output

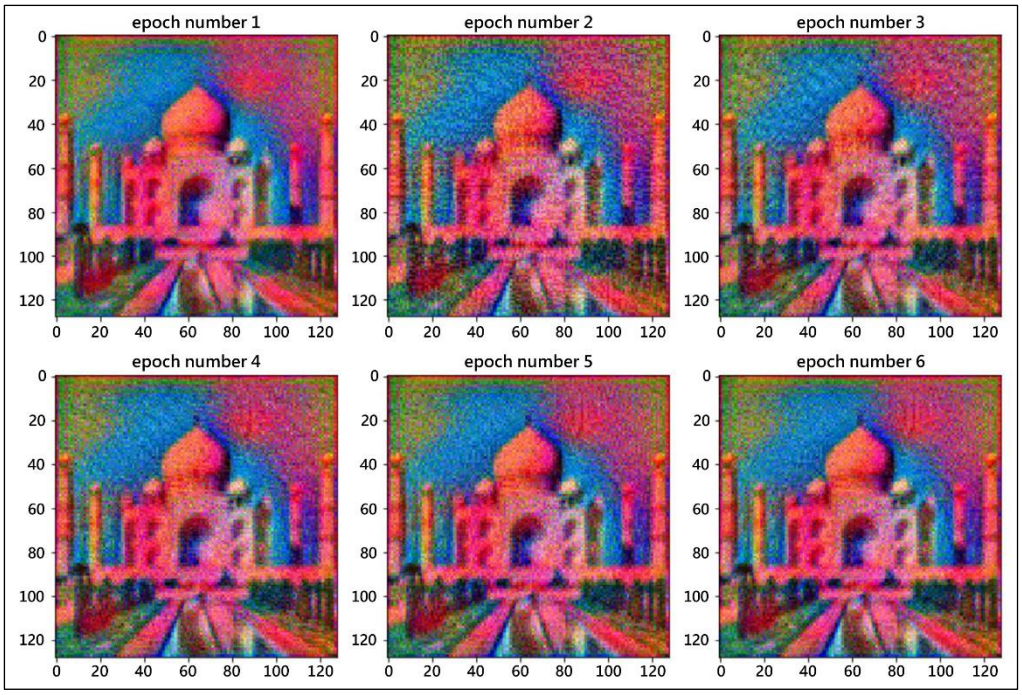

/images/style.jpg
0
500
1000
1500
2000
2500
3000
0
500
1000
1500
2000
2500
3000
/images/content.jpg
0
500
1000
1500
2000
0
500
1000
1500
2000

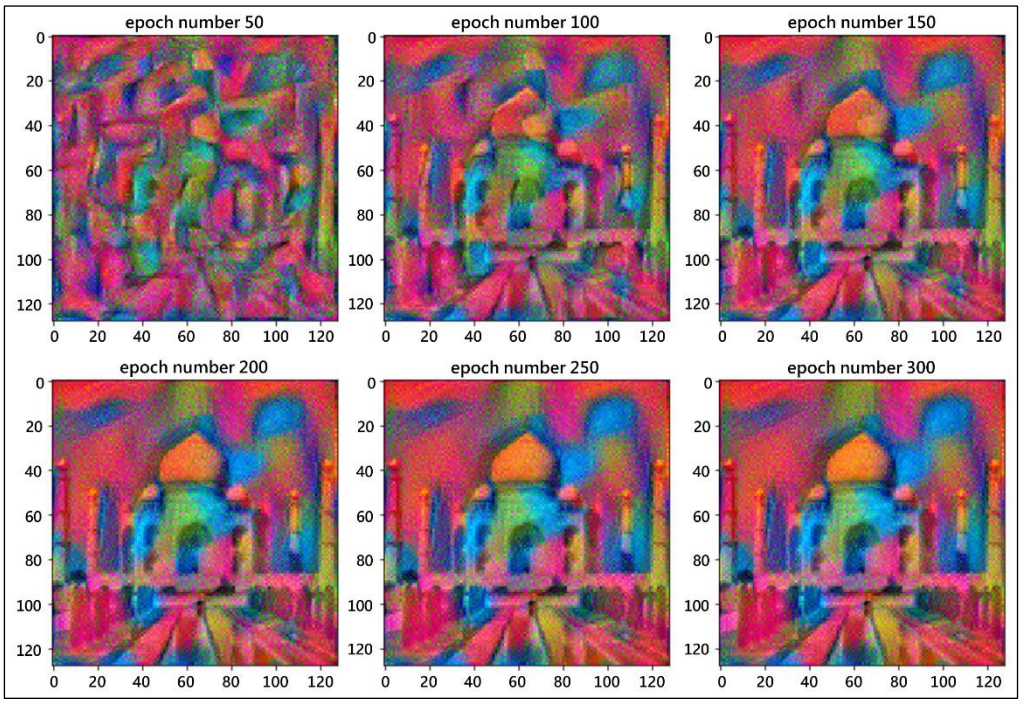

0
20
40
60
80
100
120
0 20 40 60 80 100 120

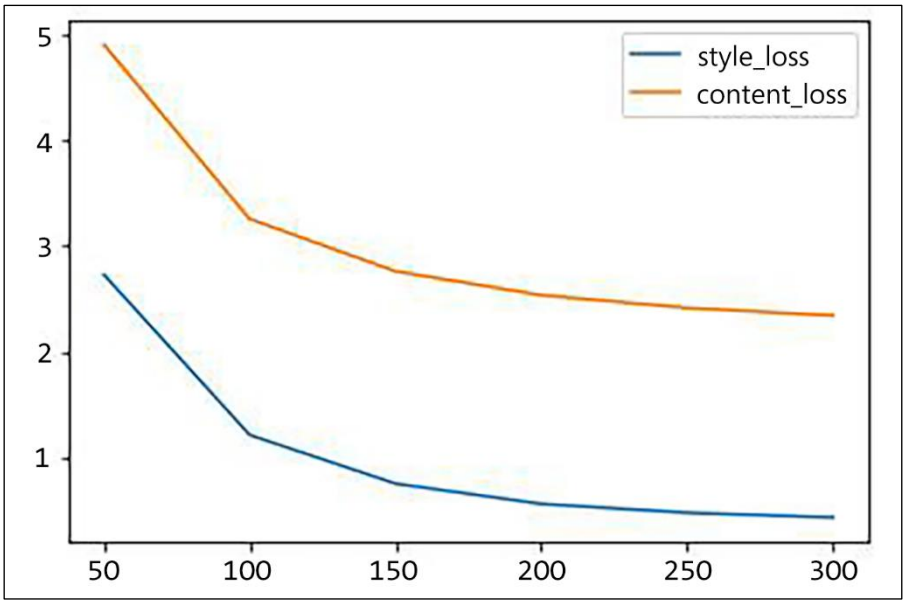

epoch number 20
epoch number 40
epoch number 60
epoch number 80
epoch number 100
epoch number 120
epoch number 140
epoch number 160
epoch number 180
0 20 40 60 80 100 120


style_loss
content_loss
25
20
15
10
5
0
20
40
60
80
100
120
140
160
180

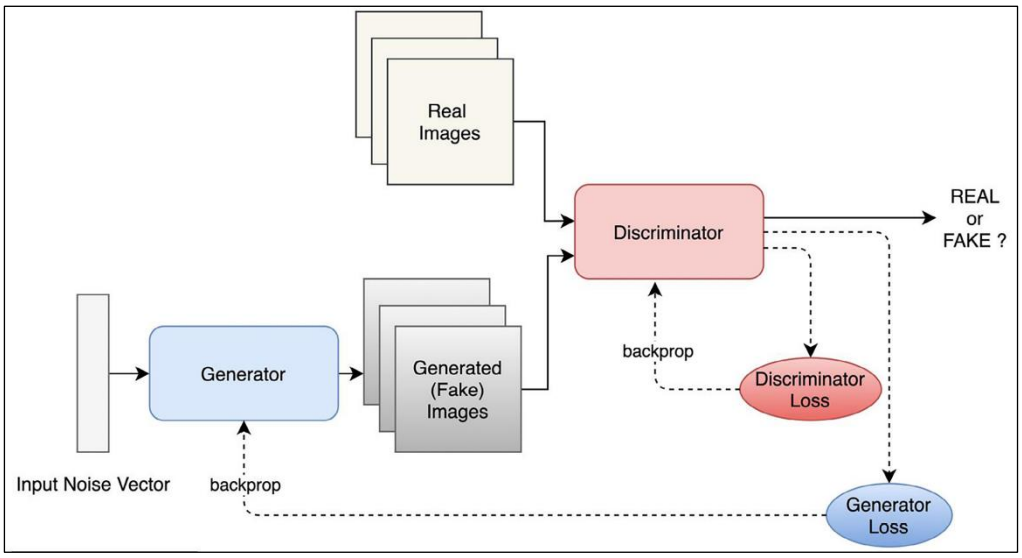

epoch number 100
epoch number 200
epoch number 300
epoch number 400
epoch number 500
epoch number 600
0
20
40
60
80
100
120

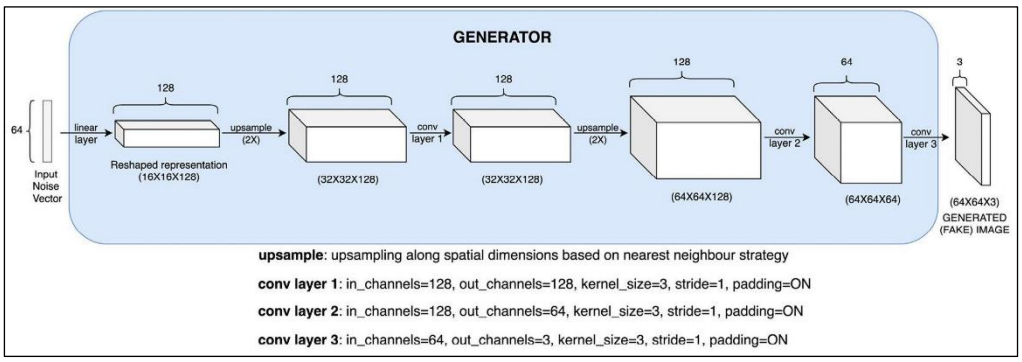

epoch number 1
epoch number 2
epoch number 3
epoch number 4
epoch number 5
epoch number 6

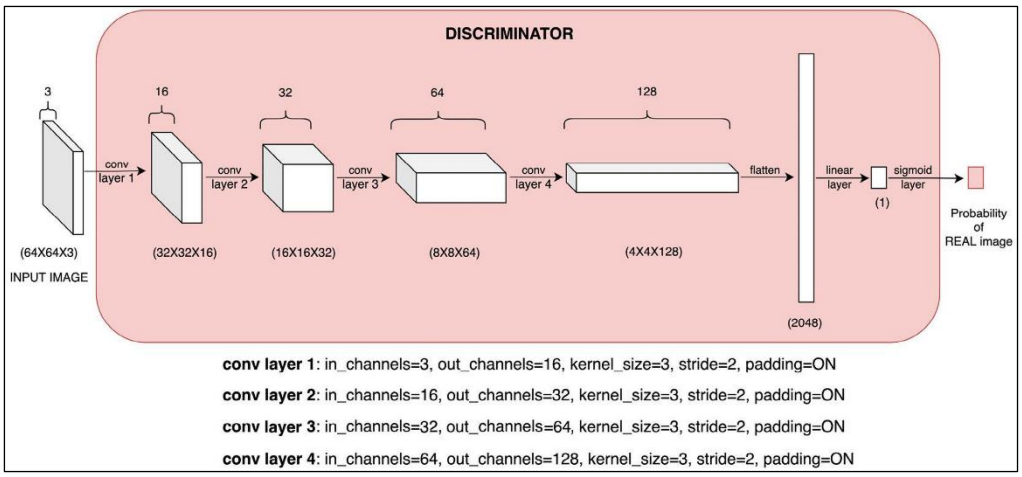

epoch number 50
epoch number 100
epoch number 150
epoch number 200
epoch number 250
epoch number 300


style_loss
content_loss
5
4
3
2
1
50
100
150
200
250
300

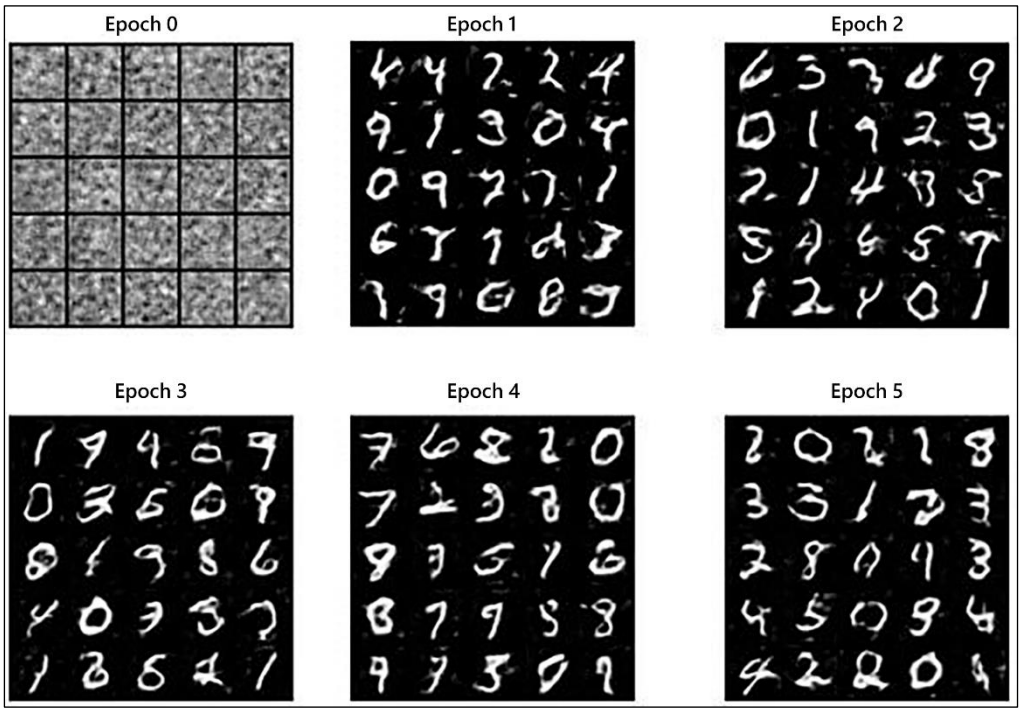
Chapter 9: Deep Convolutional GANs

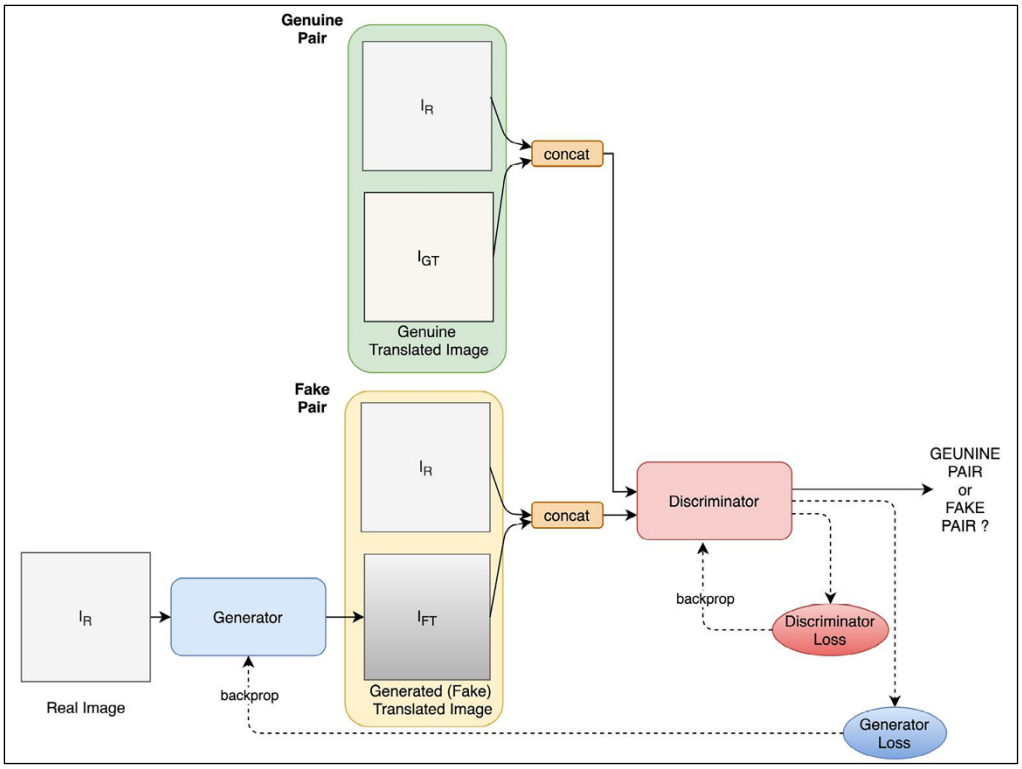


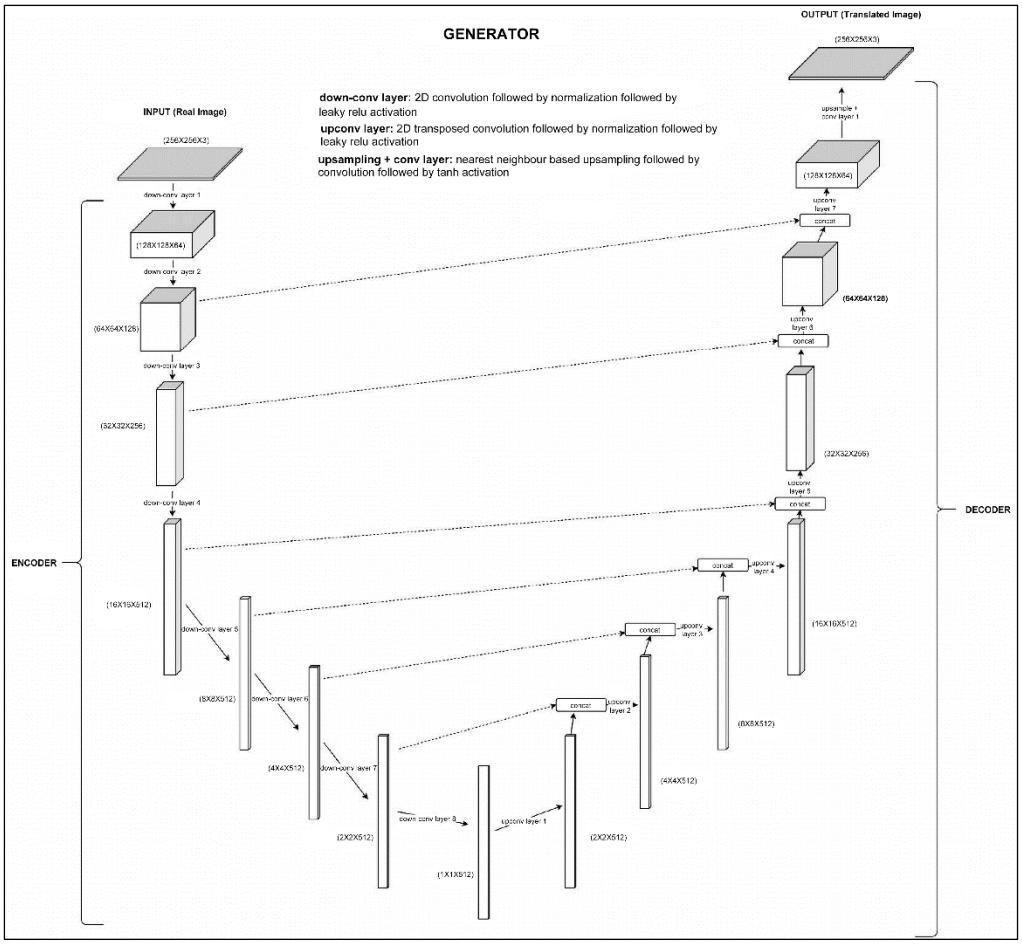


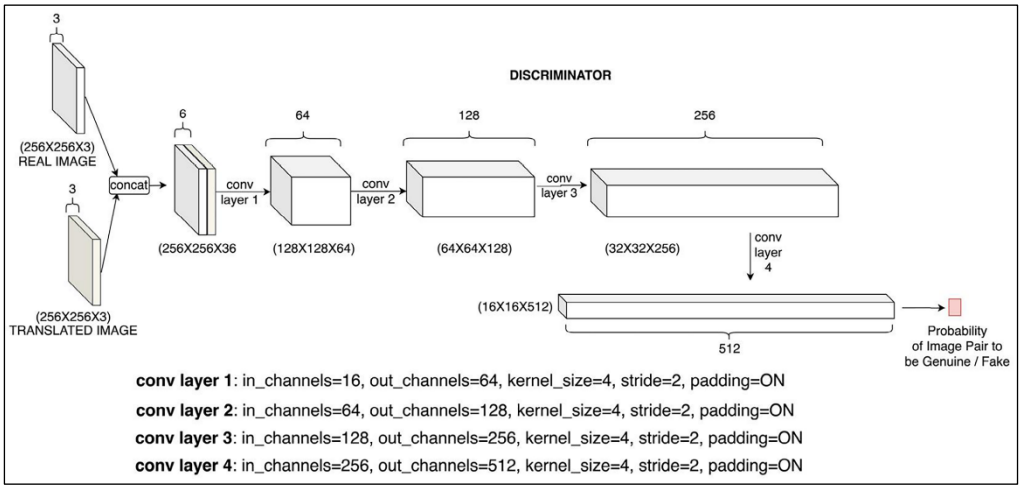





Epoch 0
Epoch 1
Epoch 2
Epoch 3
Epoch 4
Epoch 5


Genuine Pair
I_R
I_{GT}
Genuine Translated Image
concat
Fake Pair
I_R
I_{FT}
Generated (Fake) Translated Image
concat
Discriminator
GEUNINE PAIR or FAKE PAIR ?
backprop
Discriminator Loss
I_R
Real Image
Generator
backprop
Generator Loss



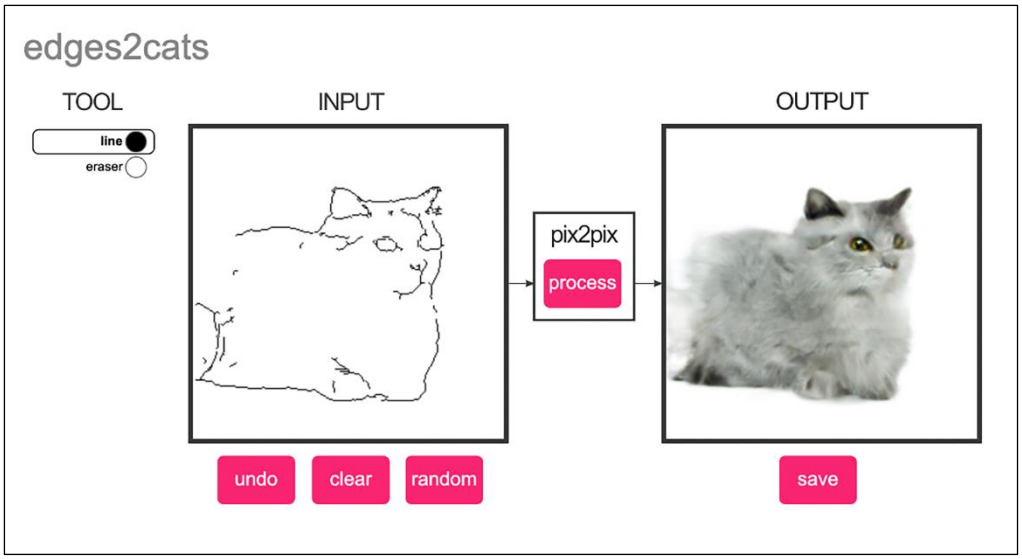


conv layer 1: in_channels=16, out_channels=64, kernel_size=4, stride=2, padding=ON

conv layer 2: in_channels=64, out_channels=128, kernel_size=4, stride=2, padding=ON

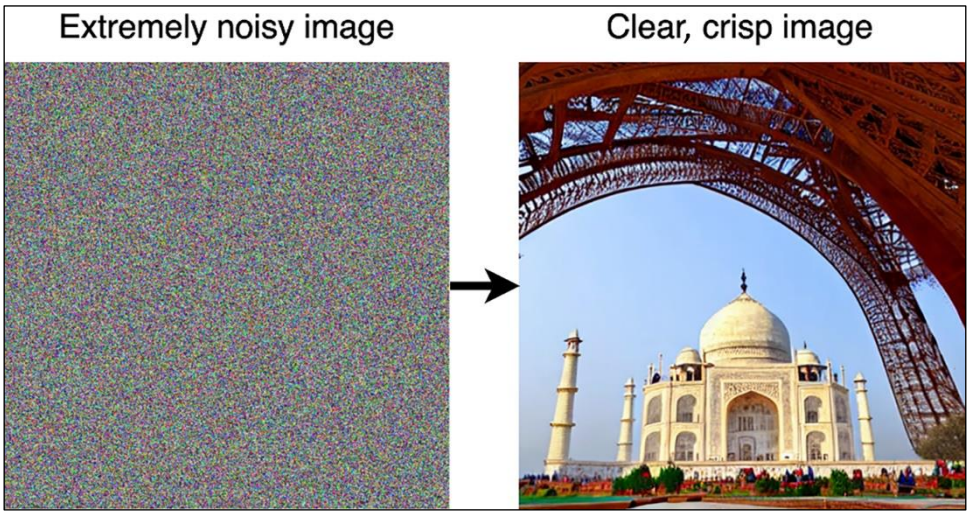
conv layer 3: in_channels=128, out_channels=256, kernel_size=4, stride=2, padding=ON

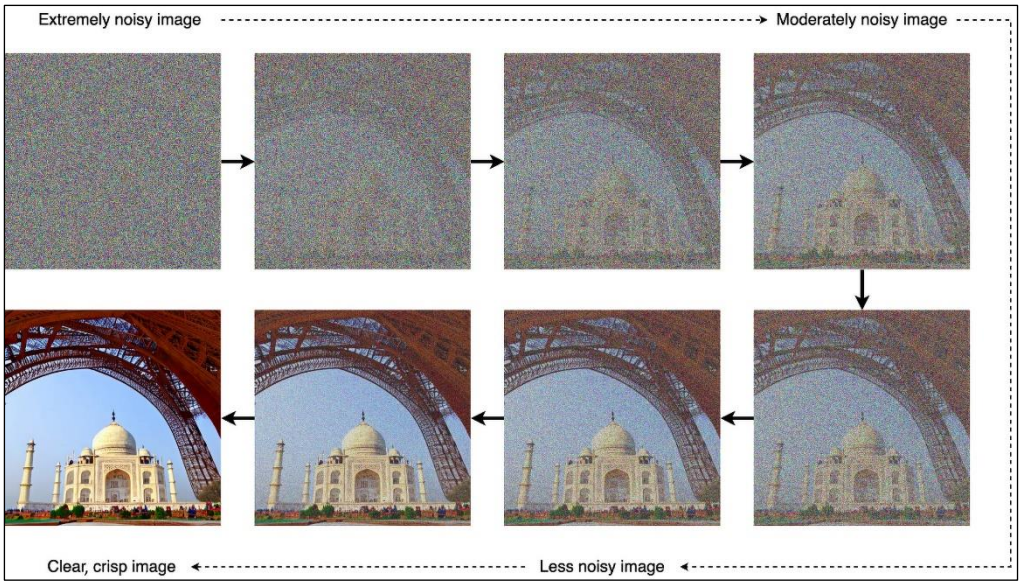
conv layer 4: in_channels=256, out_channels=512, kernel_size=4, stride=2, padding=ON

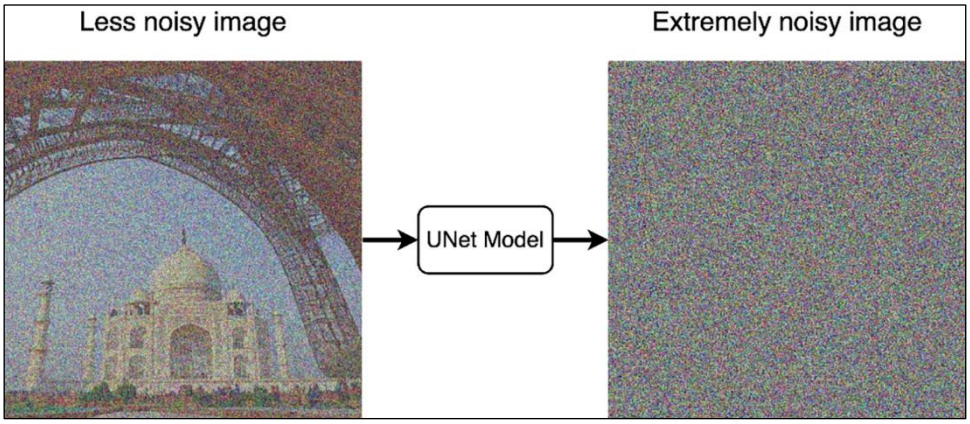

edges2cats
TOOL
line
eraser
INPUT
OUTPUT
pix2pix
process
undo
clear
random
save

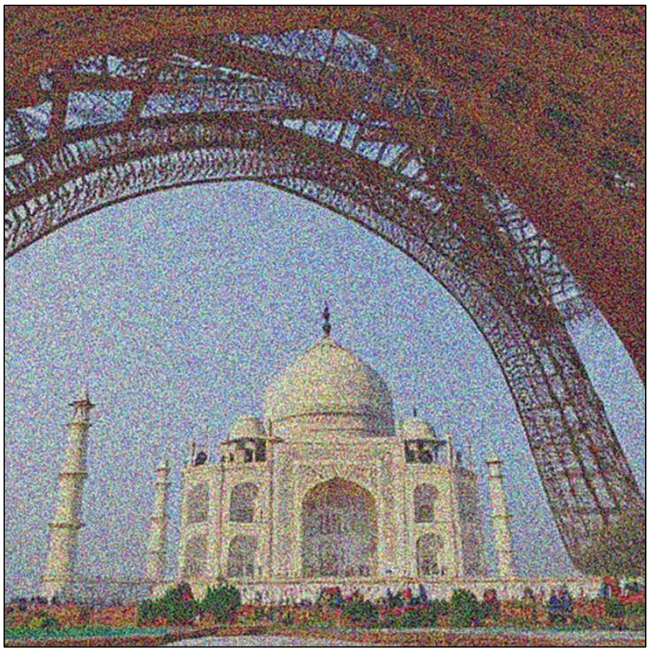
Chapter 10: Image Generation Using Diffusion

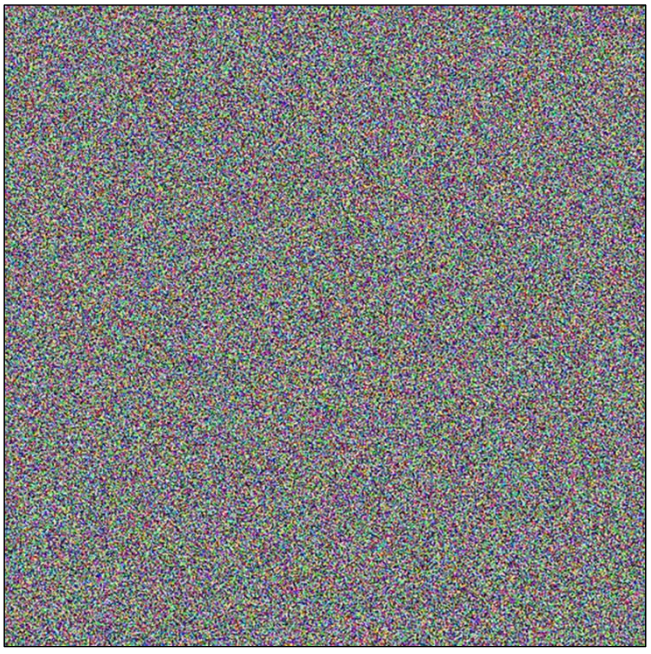


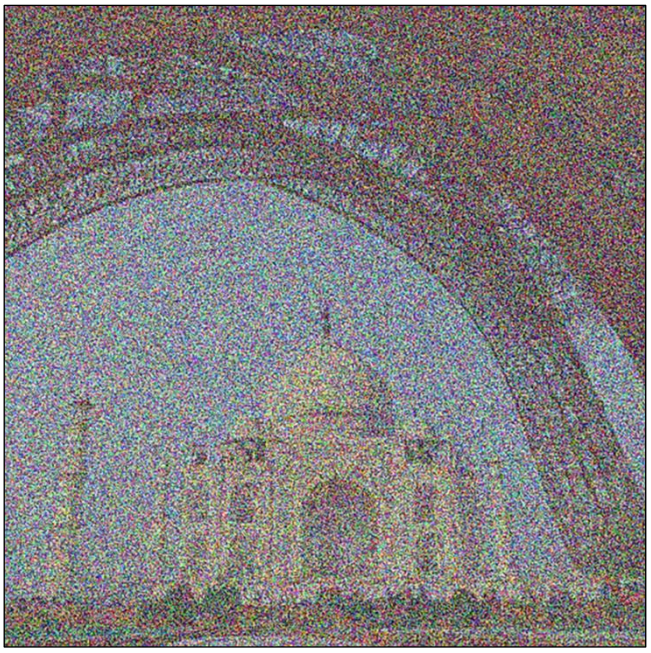


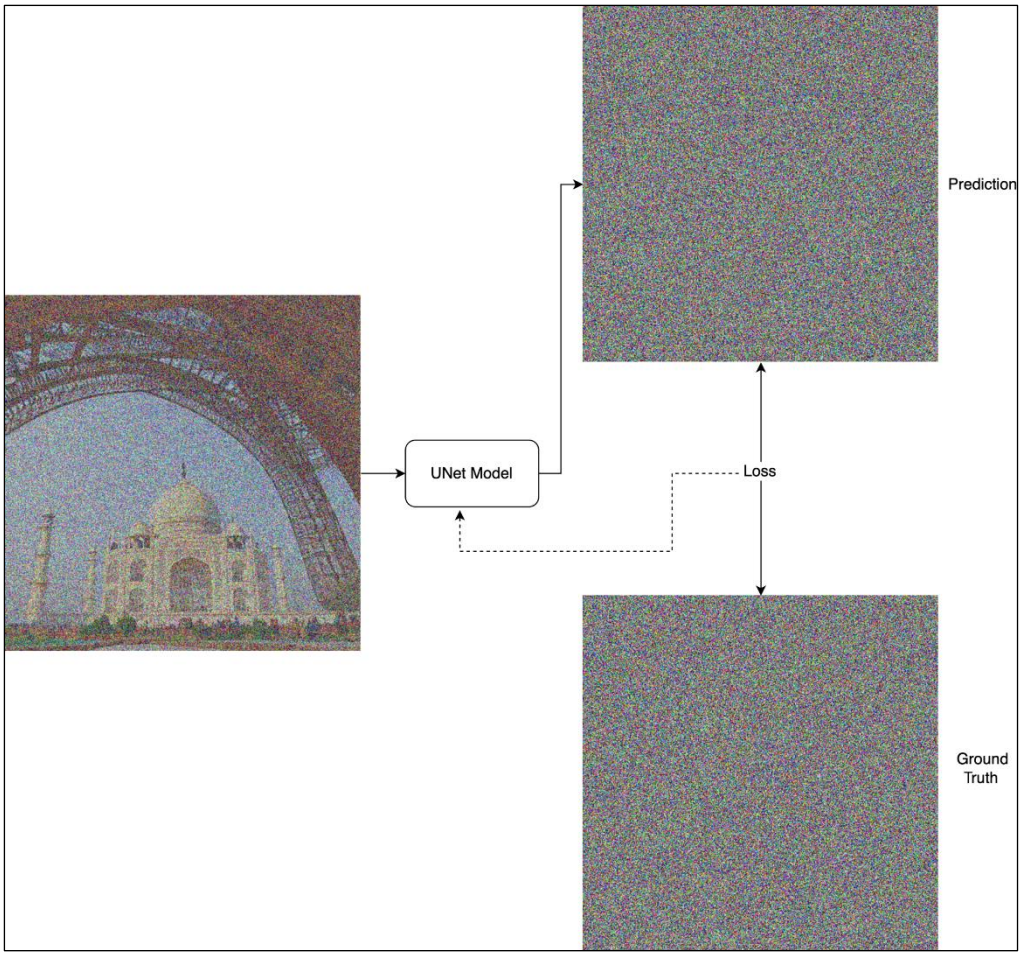

Extremely noisy image
Moderately noisy image
Clear, crisp image
Less noisy image

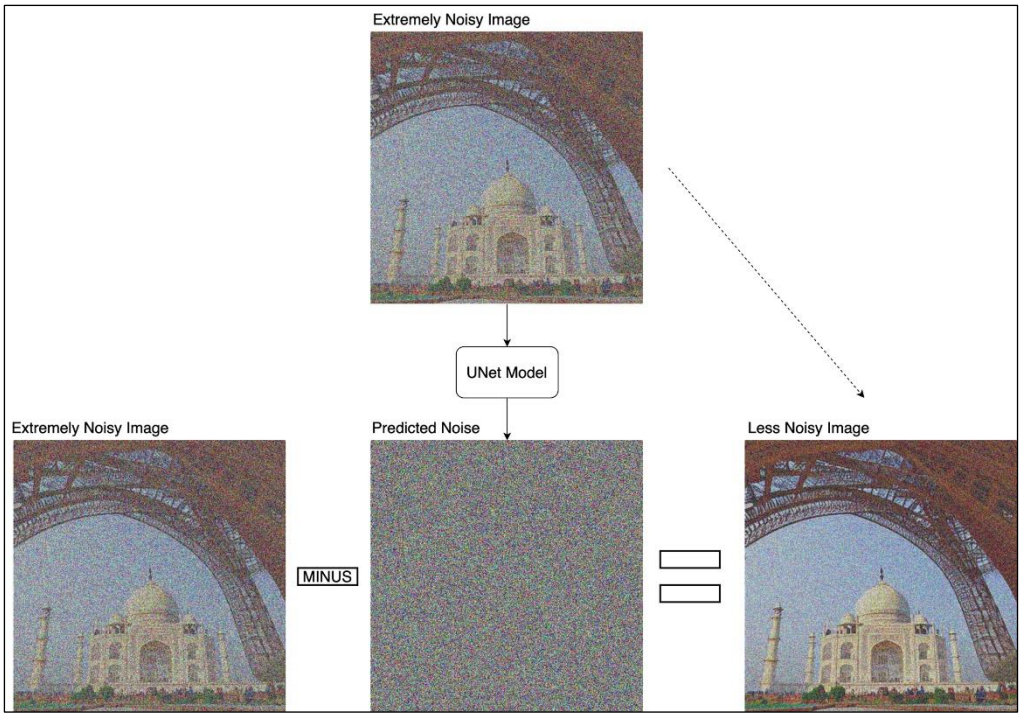

Less noisy image
Extremely noisy image
UNet Model

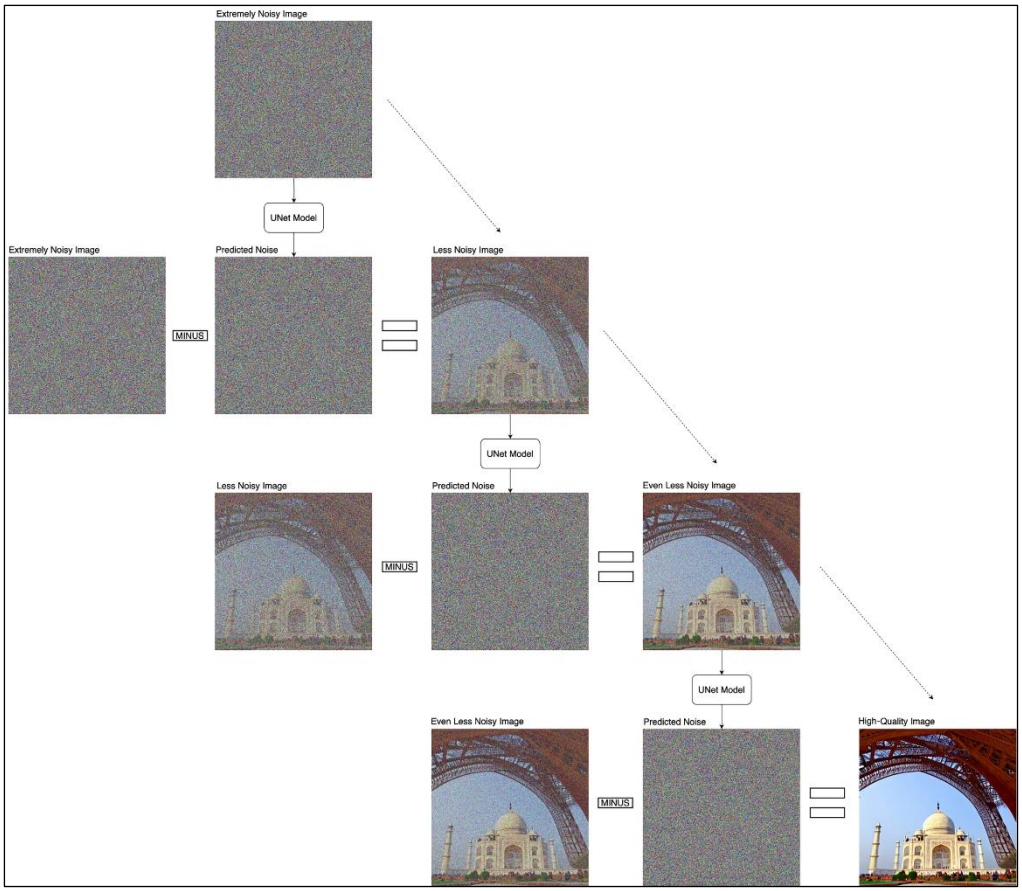


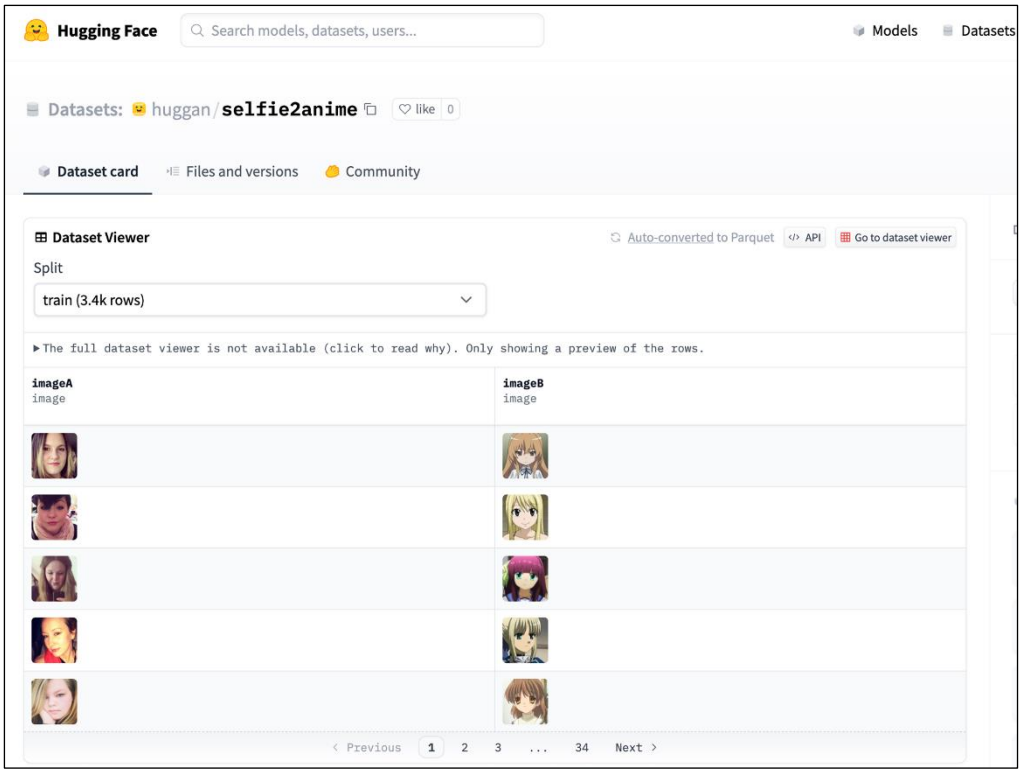





Prediction
UNet Model
Loss
Ground
Truth

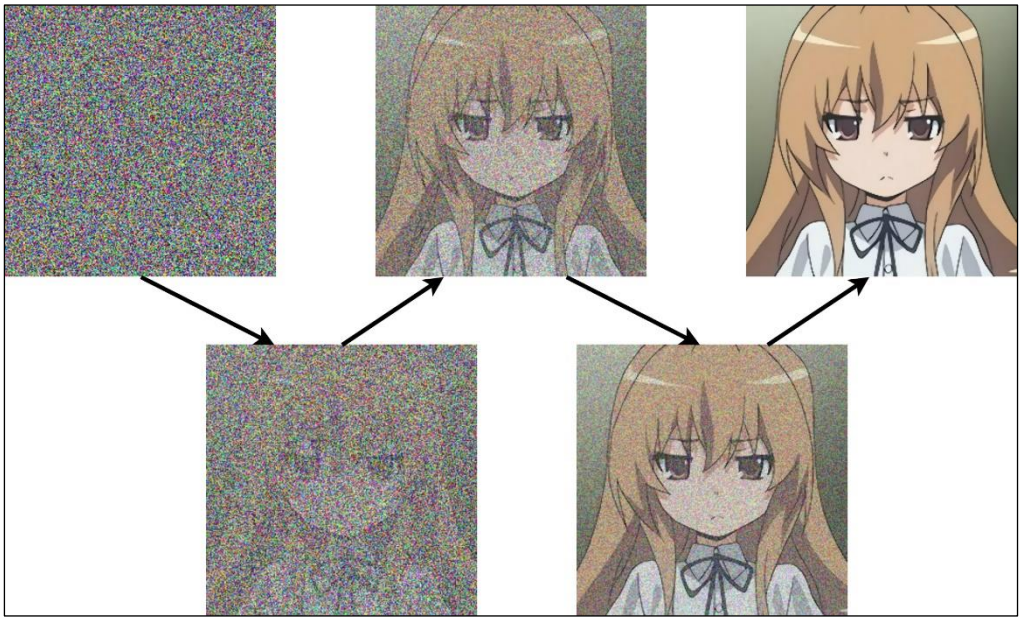

Extremely Noisy Image
UNet Model
Extremely Noisy Image
Predicted Noise
Less Noisy Image
MINUS

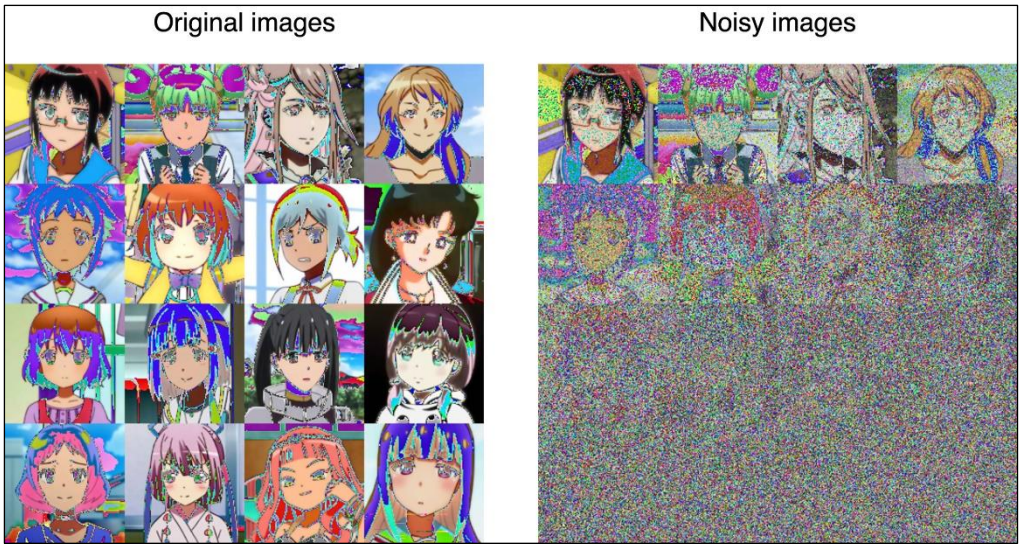

Extremely Noisy Image
UNet Model
Extremely Noisy Image
MINUS
Predicted Noise
Less Noisy Image
UNet Model
Less Noisy Image
MINUS
Predicted Noise
Even Less Noisy Image
UNet Model
Even Less Noisy Image
MINUS
Predicted Noise
High-Quality Image

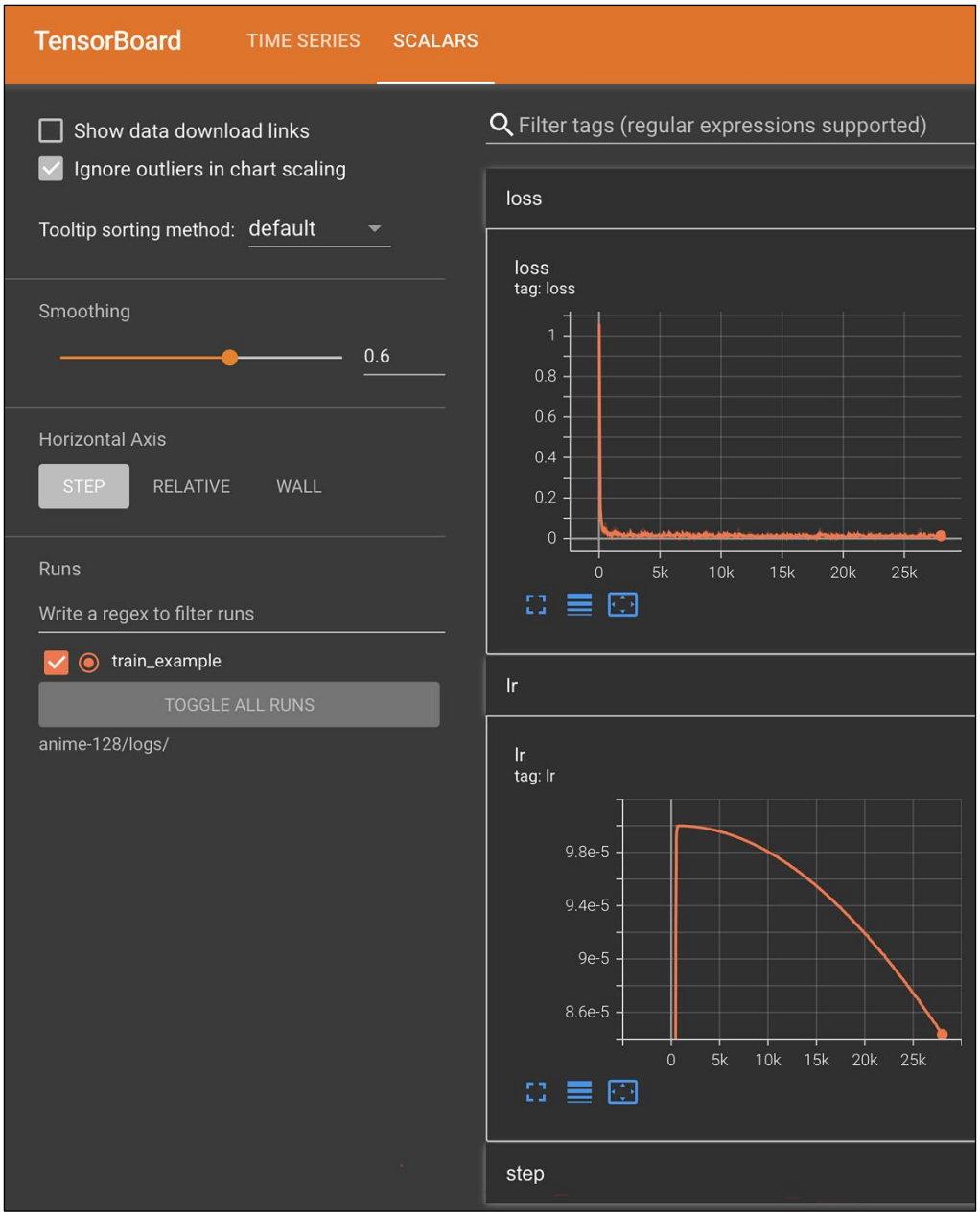

Hugging Face
Search models, datasets, users...
Models
Datasets
Datasets: huggan / selfie2anime
like 0
Dataset card
Files and versions
Community
Dataset Viewer
Auto-converted to Parquet
API
Go to dataset viewer
Split
train (3.4k rows)
The full dataset viewer is not available (click to read why). Only showing a preview of the rows.
imageA
image
imageB
image
Previous
1
2
3
...
34
Next



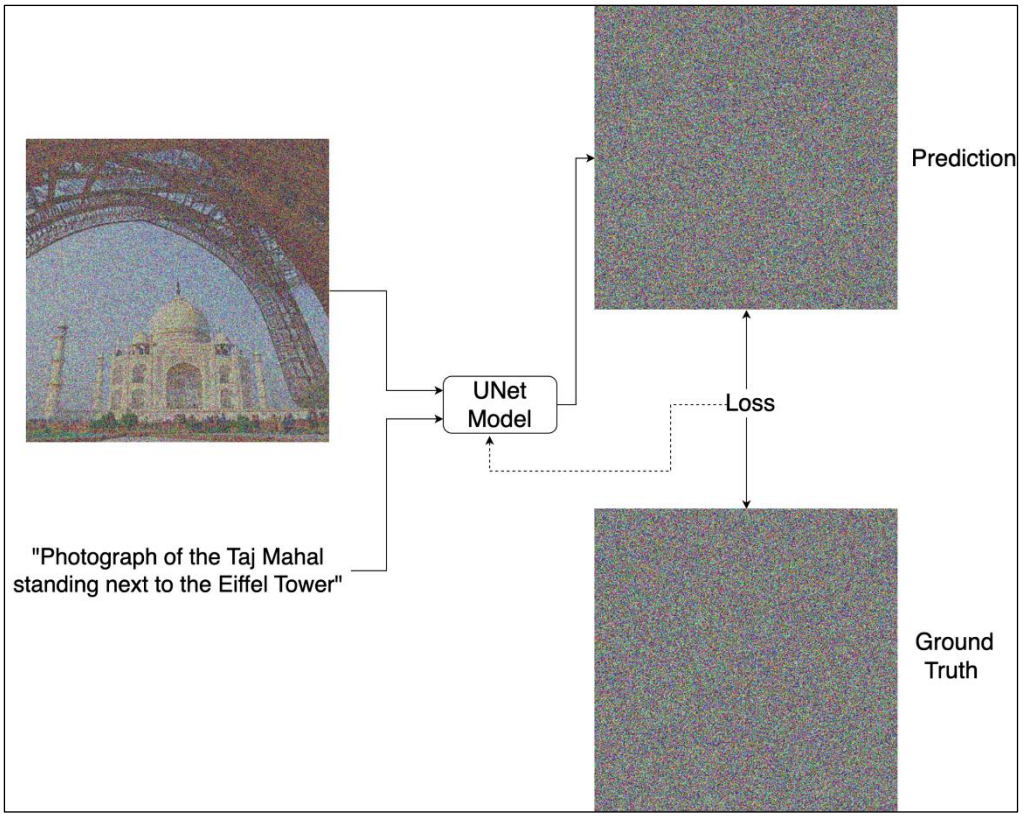


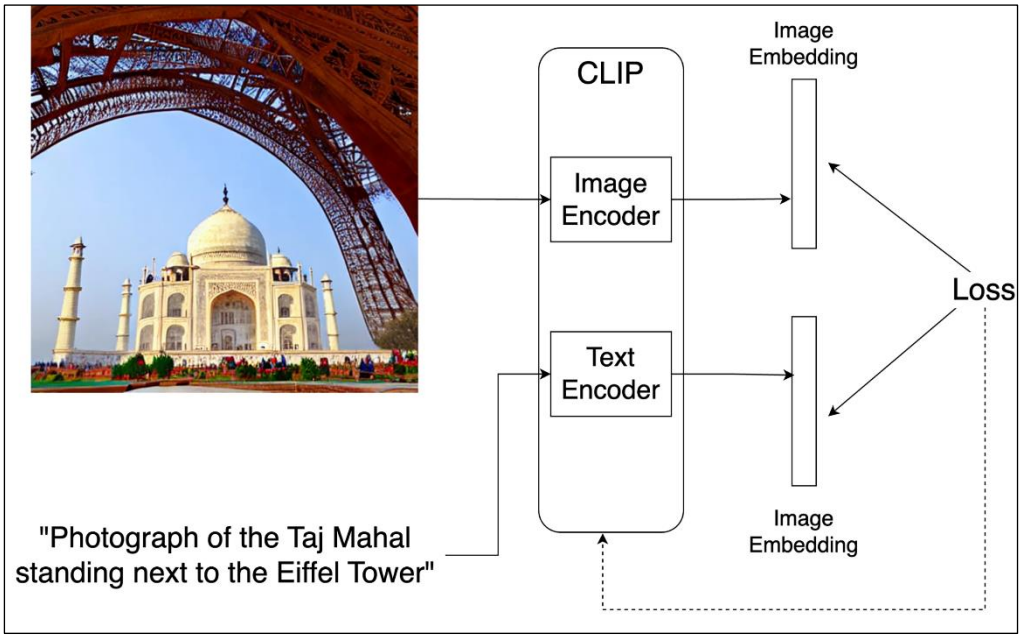


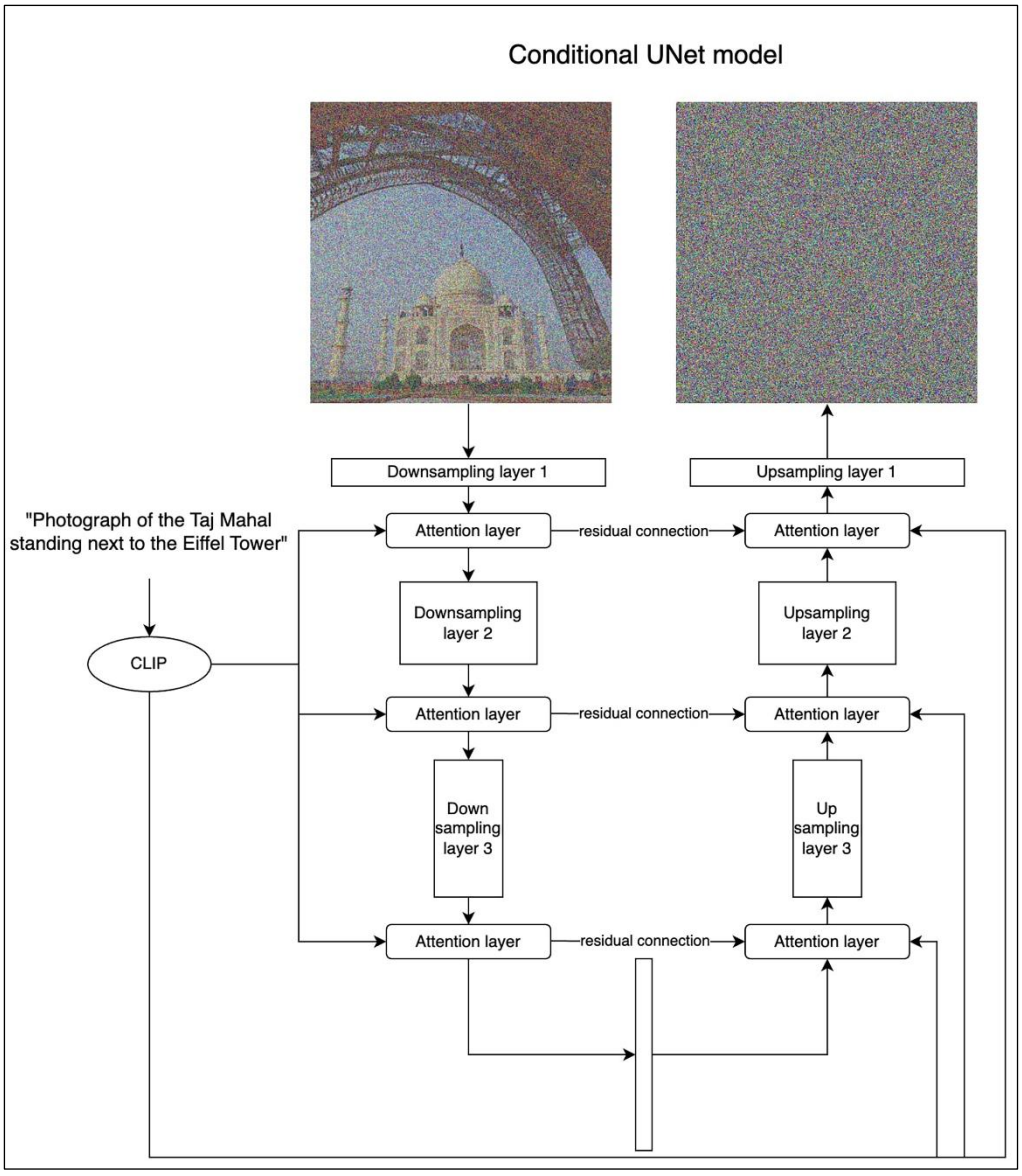

Original images
Noisy images

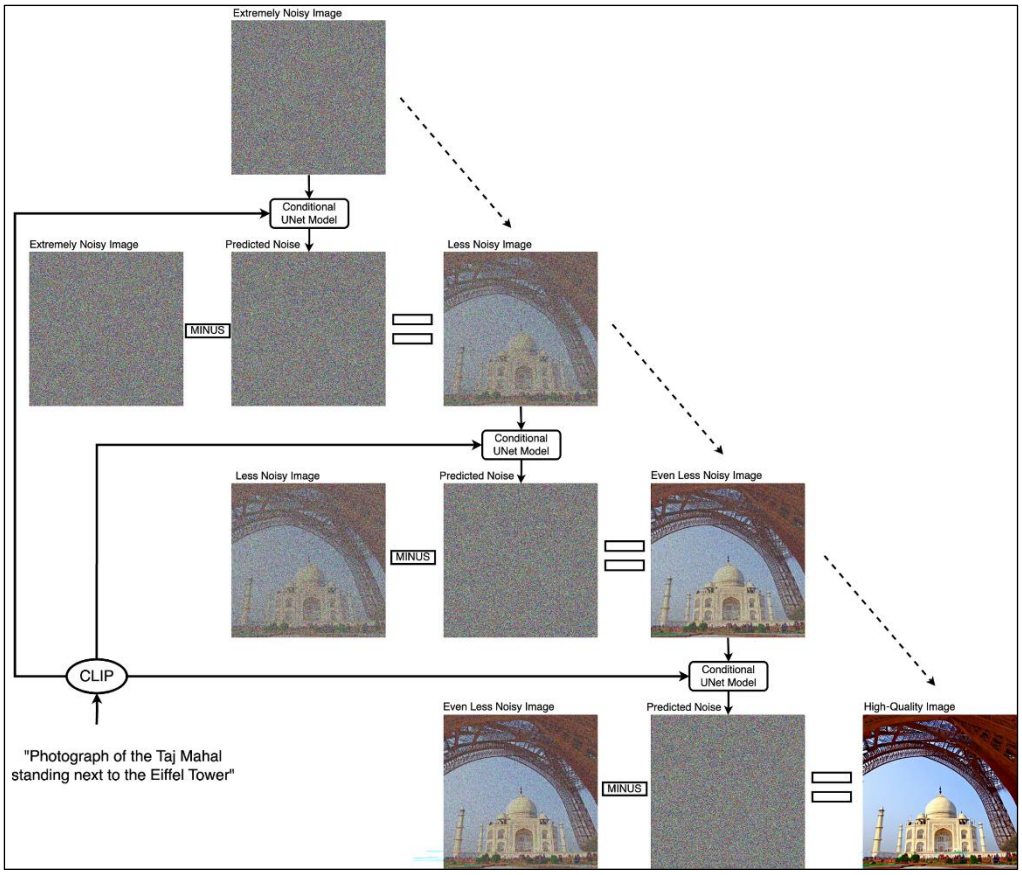

TensorBoard
TIME SERIES
SCALARS
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
train_example
TOGGLE ALL RUNS
anime-128/logs/
Filter tags (regular expressions supported)
loss
loss
tag: loss
1
0.8
0.6
0.4
0.2
0
0
5k
10k
15k
20k
25k
lr
lr
tag: lr
9.8e-5
9.4e-5
9e-5
8.6e-5
0
5k
10k
15k
20k
25k
step

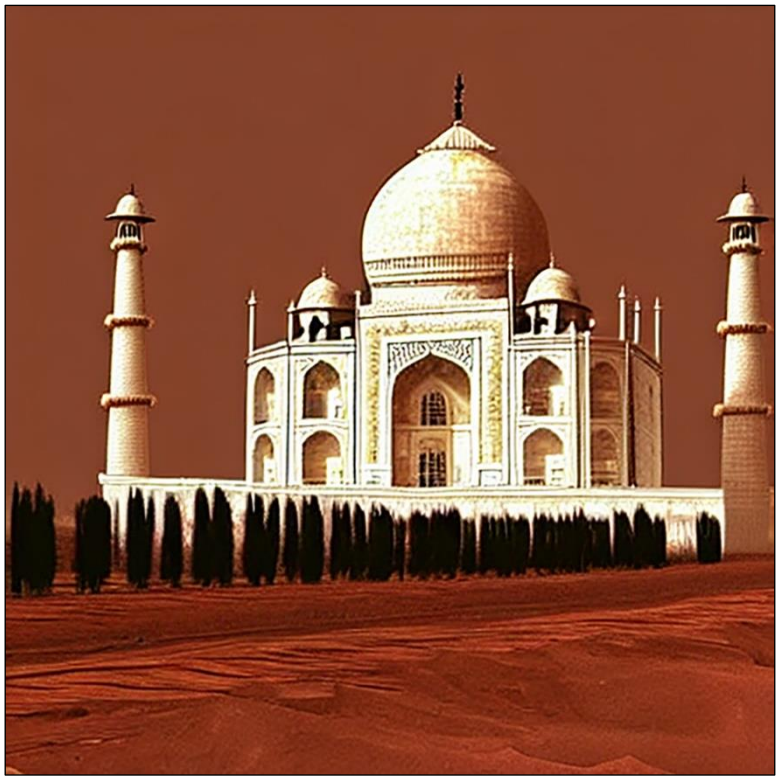

Epoch 0
Epoch 1
Epoch 2
Epoch 3
Epoch 4
Epoch 7
Epoch 8
Epoch 9
Epoch 12
Epoch 13
Epoch 17
Epoch 19


Prediction
UNet
Model
Loss
"Photograph of the Taj Mahal
standing next to the Eiffel Tower"
Ground
Truth


CLIP
Image
Embedding
Image
Encoder
Loss
Text
Encoder
Image
Embedding
"Photograph of the Taj Mahal
standing next to the Eiffel Tower"

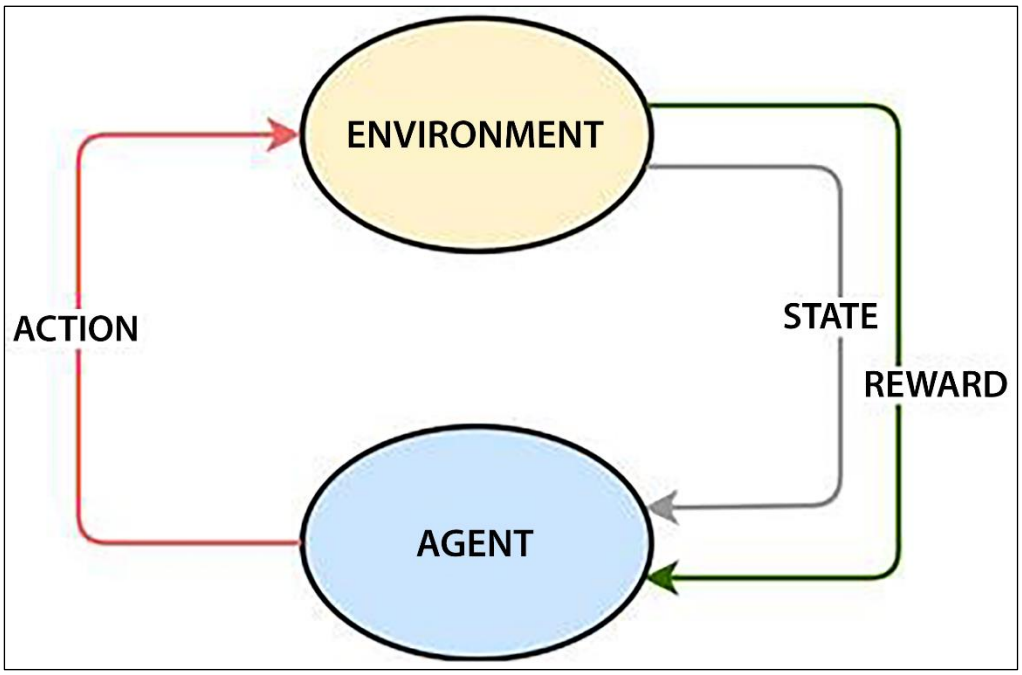

Conditional UNet model
Downsampling layer 1
Upsampling layer 1
"Photograph of the Taj Mahal standing next to the Eiffel Tower"
Attention layer
residual connection
Attention layer
Downsampling layer 2
Upsampling layer 2
CLIP
Attention layer
residual connection
Attention layer
Down sampling layer 3
Up sampling layer 3
Attention layer
residual connection
Attention layer

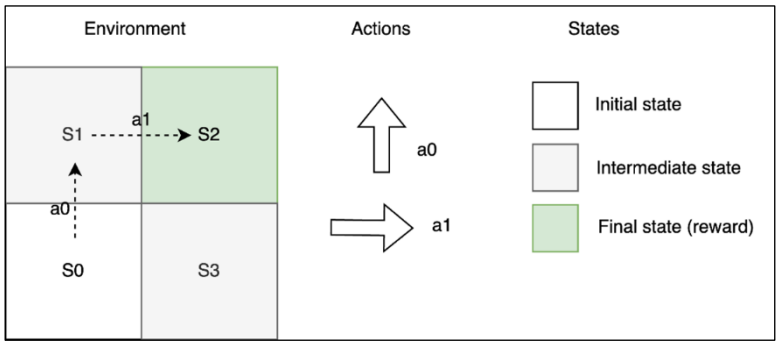

Extremely Noisy Image
Conditional UNet Model
Extremely Noisy Image
Predicted Noise
Less Noisy Image
MINUS
Conditional UNet Model
Less Noisy Image
Predicted Noise
Even Less Noisy Image
MINUS
CLIP
Conditional UNet Model
"Photograph of the Taj Mahal standing next to the Eiffel Tower"
Even Less Noisy Image
Predicted Noise
High-Quality Image
MINUS

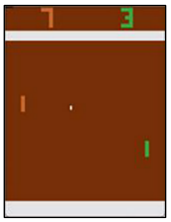


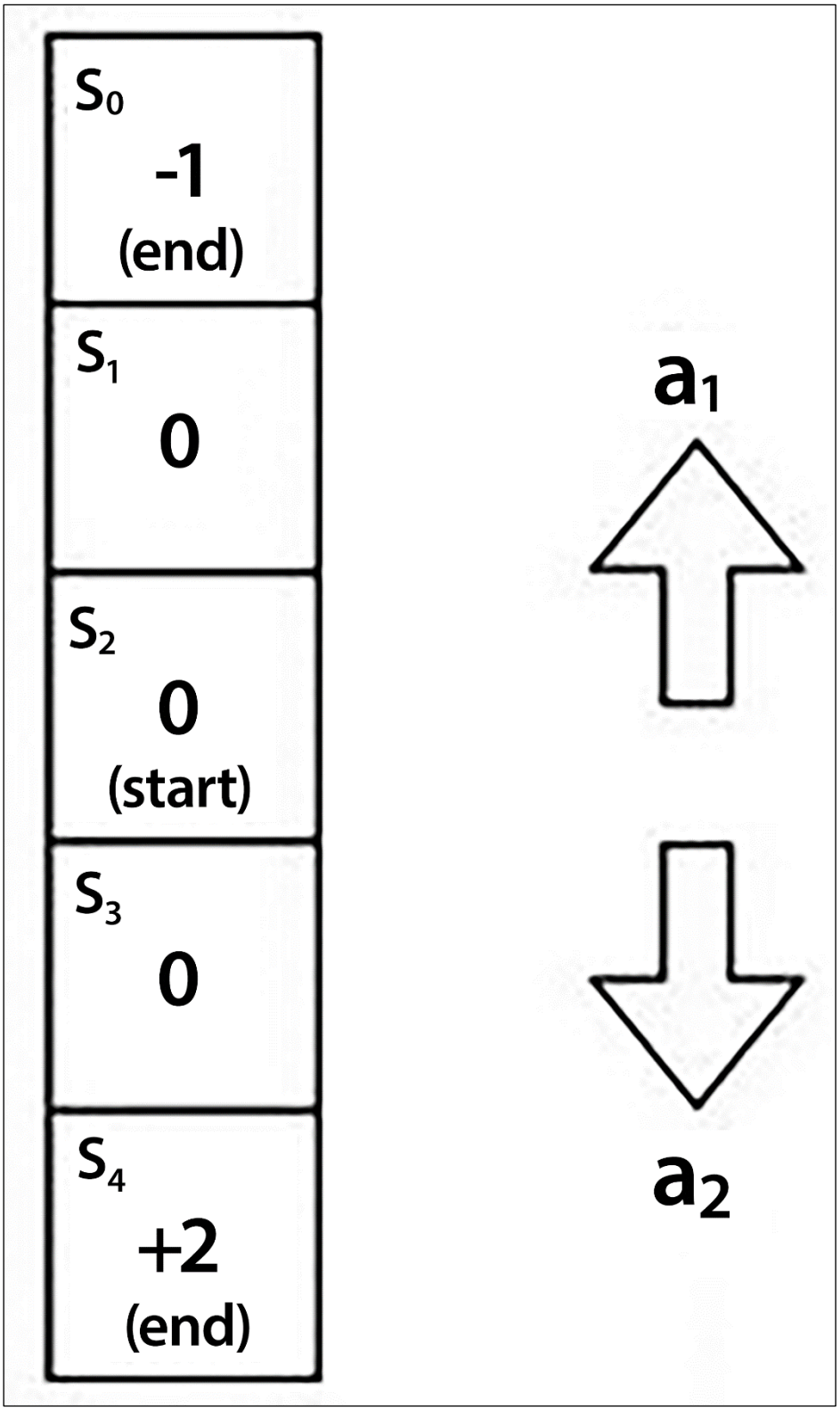


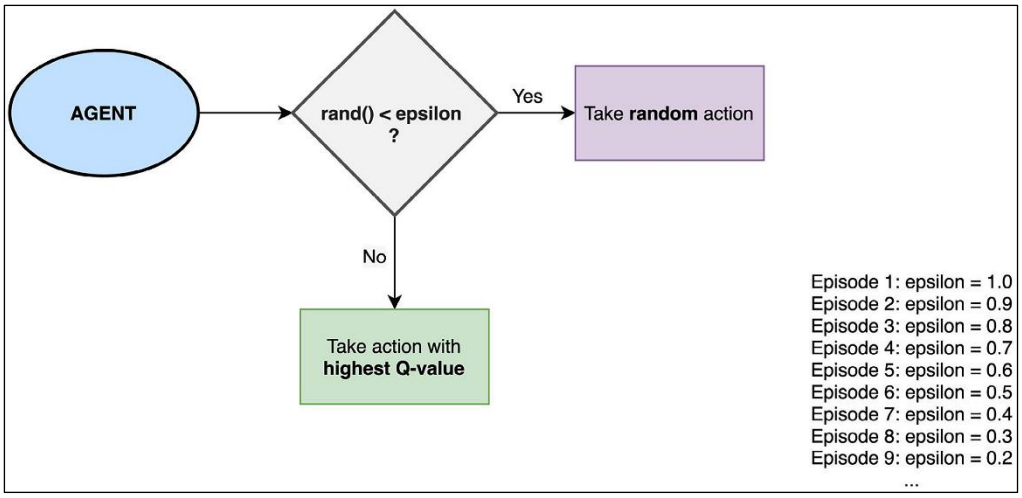
Chapter 11: Deep Reinforcement Learning

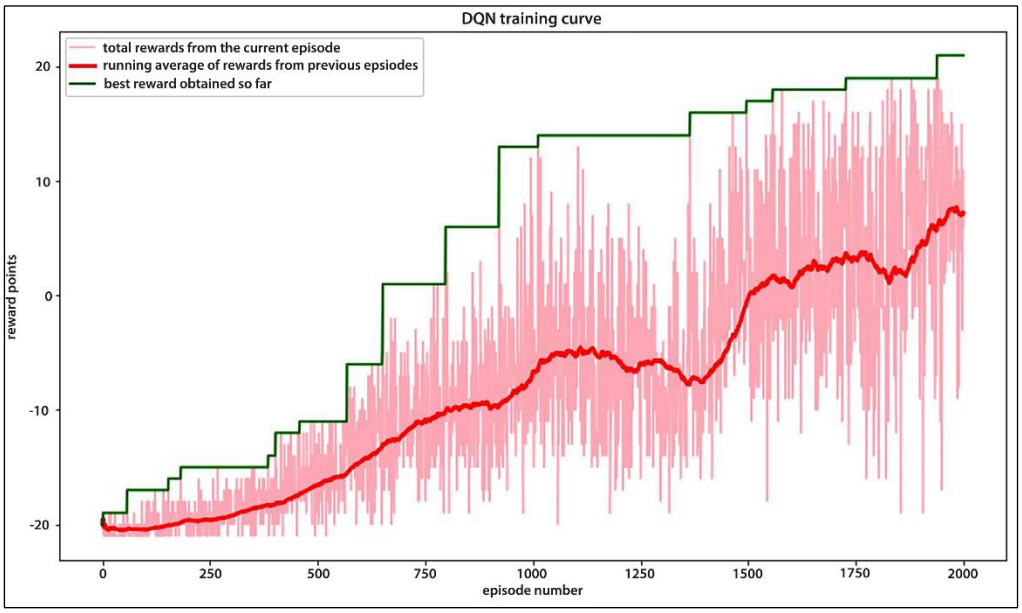


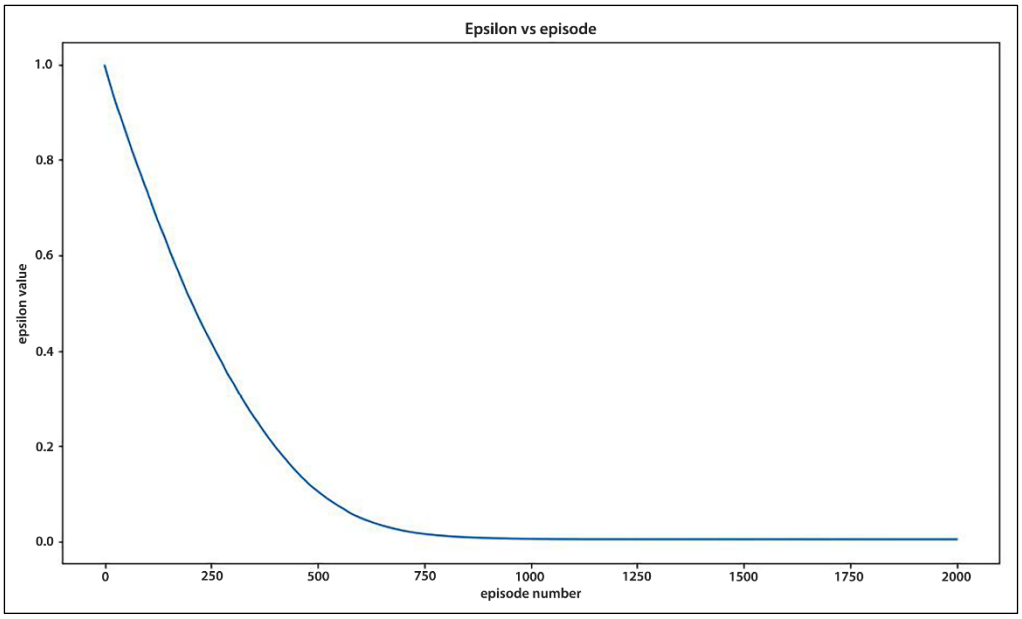




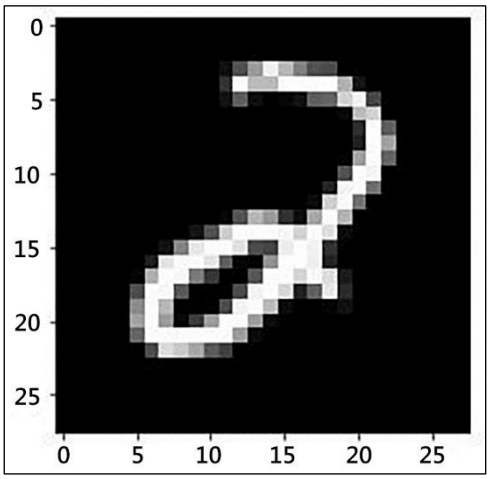

s_0
-1
(end)
s_1
0
s_2
0
(start)
s_3
0
s_4
+2
(end)
a_1
a_2


AGENT
rand() < epsilon ?
Yes
Take random action
No
Take action with highest Q-value
Episode 1: epsilon = 1.0
Episode 2: epsilon = 0.9
Episode 3: epsilon = 0.8
Episode 4: epsilon = 0.7
Episode 5: epsilon = 0.6
Episode 6: epsilon = 0.5
Episode 7: epsilon = 0.4
Episode 8: epsilon = 0.3
Episode 9: epsilon = 0.2
...


DQN training curve
total rewards from the current episode
running average of rewards from previous epsiodes
best reward obtained so far
reward points
episode number
20
10
0
-10
-20
0
250
500
750
1000
1250
1500
1750
2000


Epsilon vs episode
epsilon value
1.0
0.8
0.6
0.4
0.2
0.0
0
250
500
750
1000
1250
1500
1750
2000
episode number

Chapter 13: Operationalizing PyTorch Models into Production




```
* Serving Flask app "example" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://localhost:8898/ (Press CTRL+C to quit)
```




```
* Serving Flask app "server" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8890/ (Press CTRL+C to quit)
```

```
Sending build context to Docker daemon  7.283MB
Step 1/9 : FROM python:3.8-slim
 ---> 62297c9f4e5c
Step 2/9 : RUN apt-get -q update && apt-get -q install -y wget
 ---> Using cache
 ---> e6142d540652
Step 3/9 : COPY ./server.py ./
 ---> Using cache
 ---> cb82fb5cb2e5
Step 4/9 : COPY ./requirements.txt ./
 ---> Using cache
 ---> faa39c98f044
Step 5/9 : RUN wget -q https://github.com/PacktPublishing/Mastering-PyTorch/raw/master/Chapter10/convnet.pth
 ---> Running in 198679553bac
Removing intermediate container 198679553bac
 ---> 8bddc82ccf1e
Step 6/9 : RUN wget -q https://github.com/PacktPublishing/Mastering-PyTorch/raw/master/Chapter10/digit_image.jpg
 ---> Running in 34836205c03d
Removing intermediate container 34836205c03d
 ---> 331c8447524a
Step 7/9 : RUN pip install -r requirements.txt
 ---> Running in e4bd0692812b
Collecting torch==1.5.0
  Downloading torch-1.5.0-cp38-cp38-manylinux1_x86_64.whl (752.0 MB)
Collecting torchvision==0.6.0
  Downloading torchvision-0.6.0-cp38-cp38-manylinux1_x86_64.whl (6.6 MB)
Collecting Pillow==6.2.2
  Downloading Pillow-6.2.2-cp38-cp38-manylinux1_x86_64.whl (2.1 MB)
Collecting Flask==1.1.1
  Downloading Flask-1.1.1-py2.py3-none-any.whl (94 kB)
Collecting numpy
  Downloading numpy-1.19.5-cp38-cp38-manylinux2010_x86_64.whl (14.9 MB)
Collecting future
  Downloading future-0.18.2.tar.gz (829 kB)
Collecting Werkzeug>=0.15
  Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
Collecting itsdangerous>=0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2>=2.10.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
Collecting click>=5.1
  Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
Collecting MarkupSafe>=0.23
  Downloading MarkupSafe-1.1.1-cp38-cp38-manylinux1_x86_64.whl (32 kB)
Building wheels for collected packages: future
  Building wheel for future (setup.py): started
  Building wheel for future (setup.py): finished with status 'done'
  Created wheel for future: filename=future-0.18.2-py3-none-any.whl size=491059 sha256=726027831b4150f39c497dee6280cf48539056b1b80d80fbf113330e7ea5af0d
  Stored in directory: /root/.cache/pip/wheels/8e/70/28/3d6ccd6e315f65f245da085482a2e1c7d14b90b30f239e2cf4
Successfully built future
Installing collected packages: numpy, future, torch, Pillow, torchvision, Werkzeug, itsdangerous, MarkupSafe, Jinja2, click, Flask
Successfully installed Flask-1.1.1 Jinja2-2.11.2 MarkupSafe-1.1.1 Pillow-6.2.2 Werkzeug-1.0.1 click-7.1.2 future-0.18.2 itsdangerous-1.1.0 numpy-1.19.5 torch-1.5.0 torchvision-0.6.0
WARNING: You are using pip version 20.2.3; however, version 20.3.3 is available.
You should consider upgrading via the '/usr/local/bin/python -m pip install --upgrade pip' command.
Removing intermediate container e4bd0692812b
 ---> d7ecf4681868
Step 8/9 : USER root
 ---> Running in 90dac164d1bc
Removing intermediate container 90dac164d1bc
 ---> e4488fab0082
Step 9/9 : ENTRYPOINT ["python", "server.py"]
 ---> Running in a1d55f33a99f
Removing intermediate container a1d55f33a99f
 ---> 2de7d75aae1b
Successfully built 2de7d75aae1b
Successfully tagged digit_recognizer:latest
```

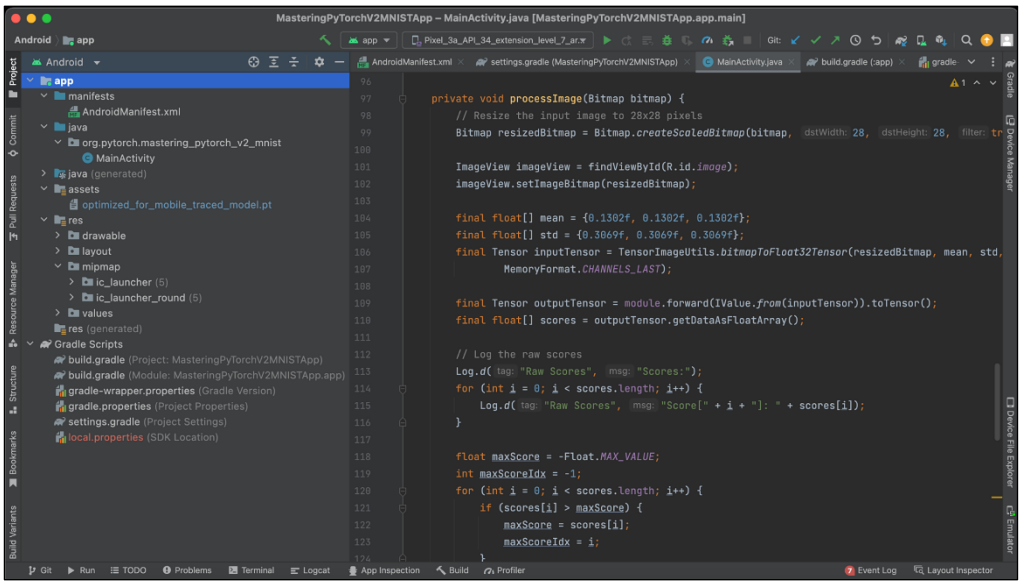
```
* Serving Flask app "server" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8890/ (Press CTRL+C to quit)
```

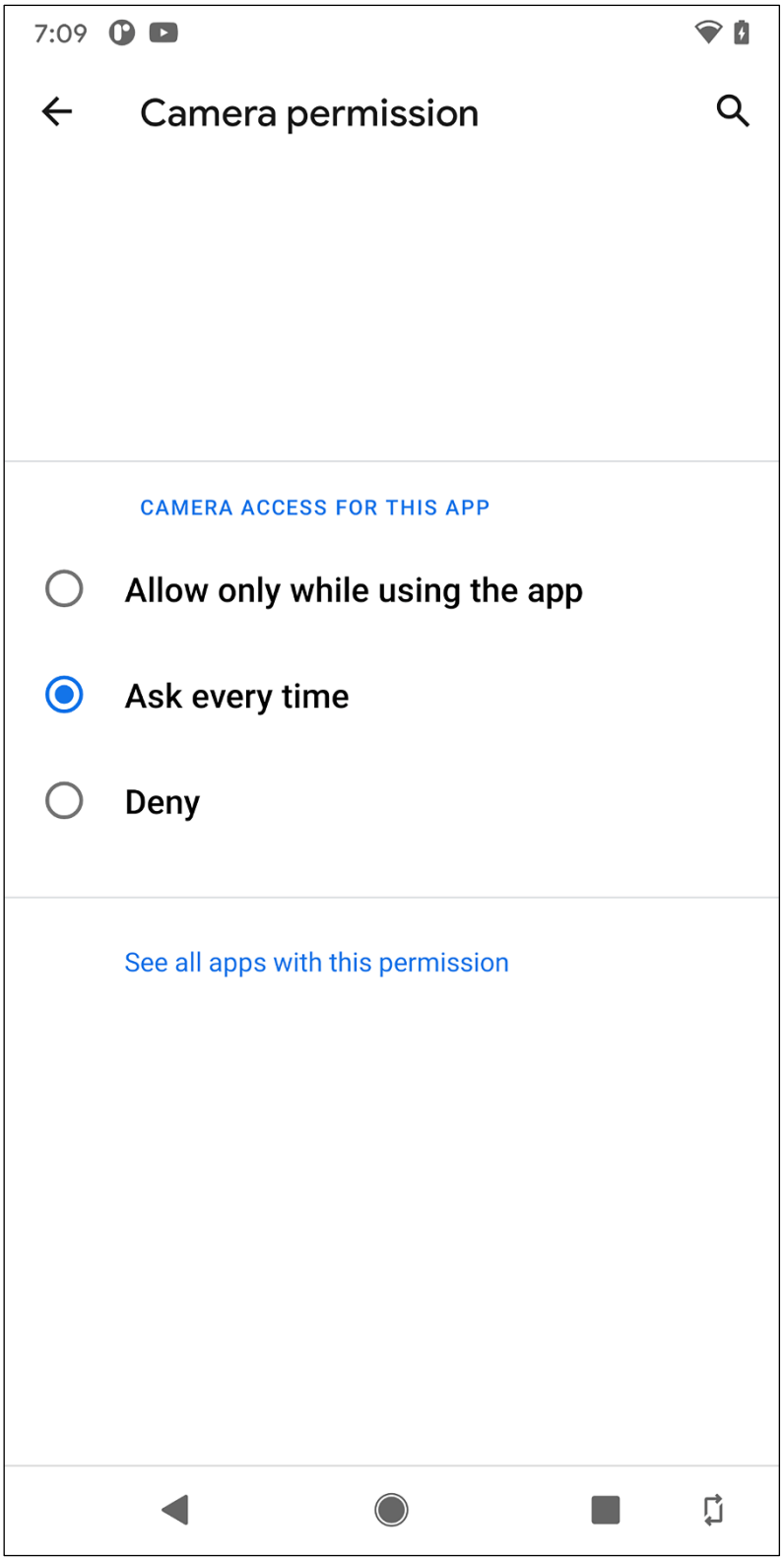
```
-- The C compiler identification is AppleClang 13.1.6.13160021
-- The CXX compiler identification is AppleClang 13.1.6.13160021
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /Library/Developer/CommandLineTools/usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /Library/Developer/CommandLineTools/usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Success
-- Found Threads: TRUE
-- MKL_ARCH: None, set to ` intel64` by default
-- MKL_ROOT /Users/ashish.jha/opt/anaconda3/envs/mastering_pytorch_7_chaps
-- MKL_LINK: None, set to ` dynamic` by default
-- MKL_INTERFACE_FULL: None, set to ` intel_ilp64` by default
-- MKL_THREADING: None, set to ` intel_thread` by default
-- MKL_MPI: None, set to ` mpich` by default
CMake Warning at /Users/ashish.jha/opt/anaconda3/envs/mastering_pytorch_7_chaps/lib/python3.9/site-packages/torch/share/cmake/Torch/TorchConfig.cmake:22 (message):
  static library kineto_LIBRARY-NOTFOUND not found.
Call Stack (most recent call first):
  /Users/ashish.jha/opt/anaconda3/envs/mastering_pytorch_7_chaps/lib/python3.9/site-packages/torch/share/cmake/Torch/TorchConfig.cmake:127 (append_torchlib_if_found)
  CMakeLists.txt:3 (find_package)

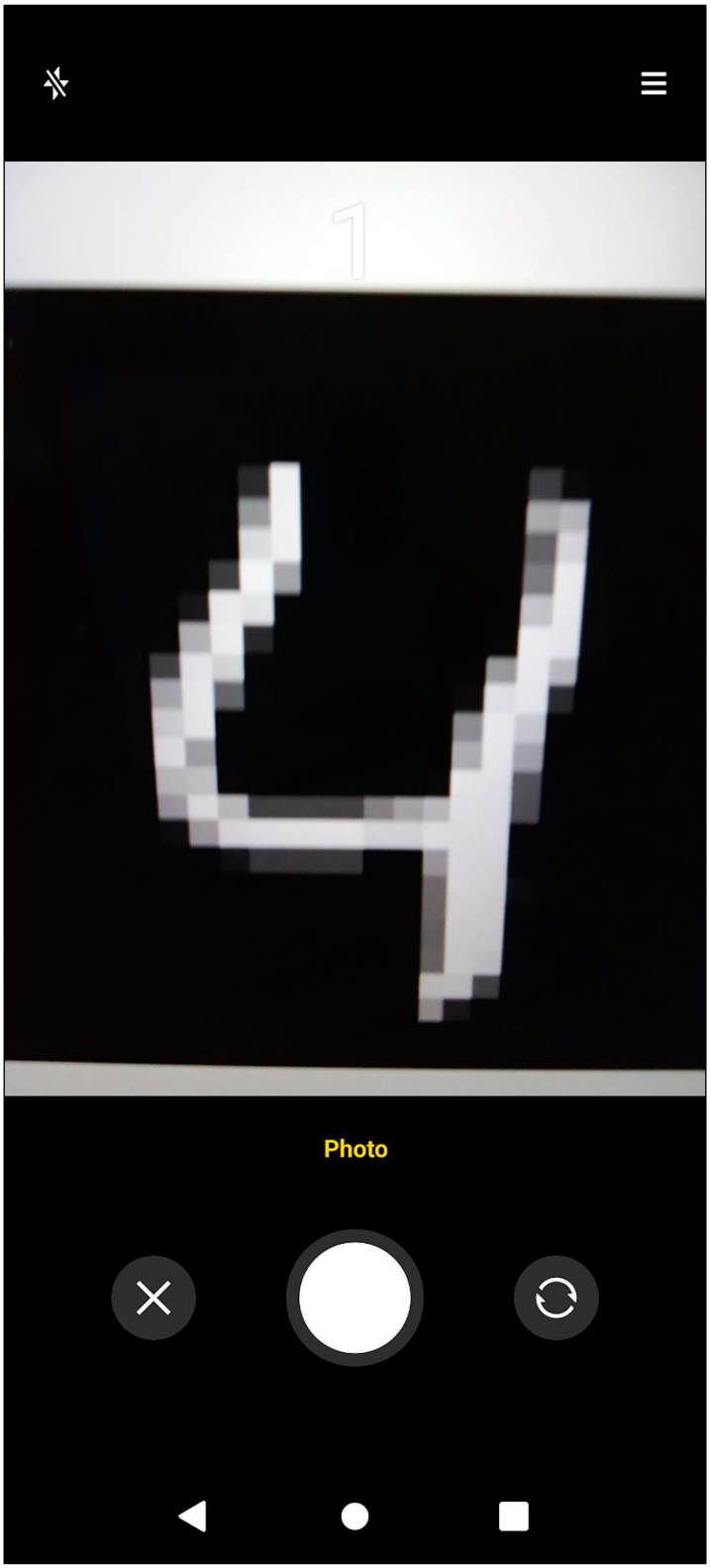

-- Found Torch: /Users/ashish.jha/opt/anaconda3/envs/mastering_pytorch_7_chaps/lib/python3.9/site-packages/torch/lib/libtorch.dylib
-- Found OpenCV: /Users/ashish.jha/code/personal/Mastering-PyTorch/Chapter13/cpp_convnet/build_opencv (found version "4.6.0")
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/ashish.jha/code/personal/Mastering-PyTorch/Chapter13/cpp_convnet/build
```

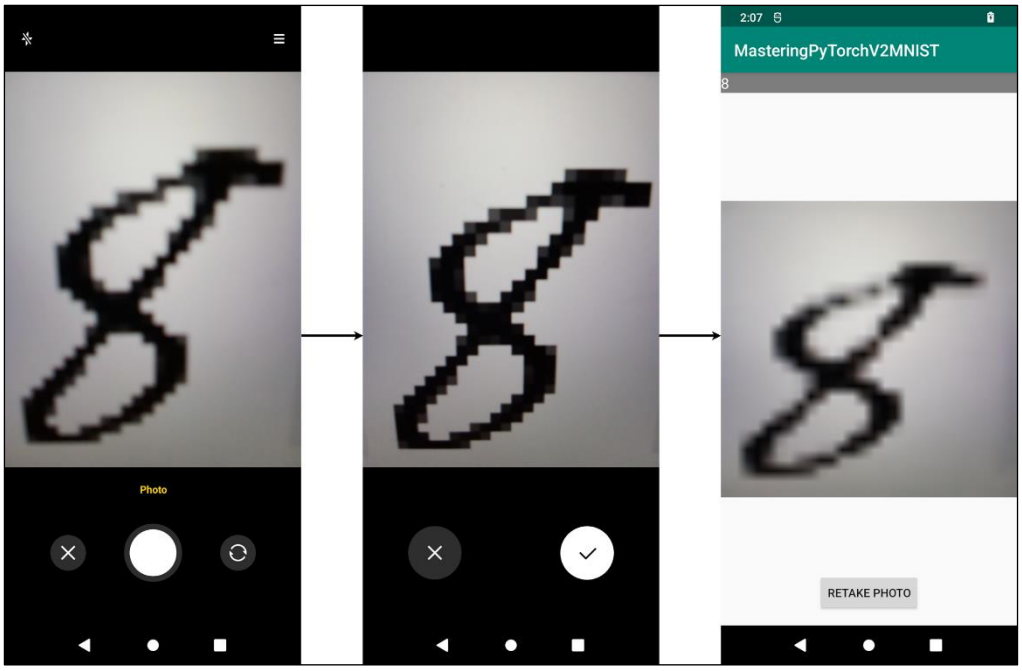
```
Scanning dependencies of target cpp_convnet
[ 50%] Building CXX object CMakeFiles/cpp_convnet.dir/cpp_convnet.cpp.o
[100%] Linking CXX executable cpp_convnet
[100%] Built target cpp_convnet
```

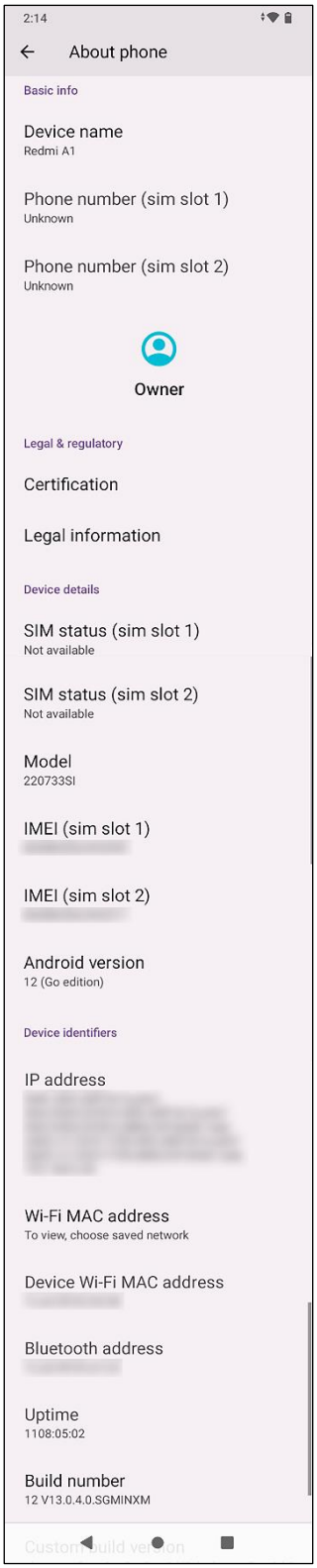
Chapter 14: PyTorch on Mobile Devices

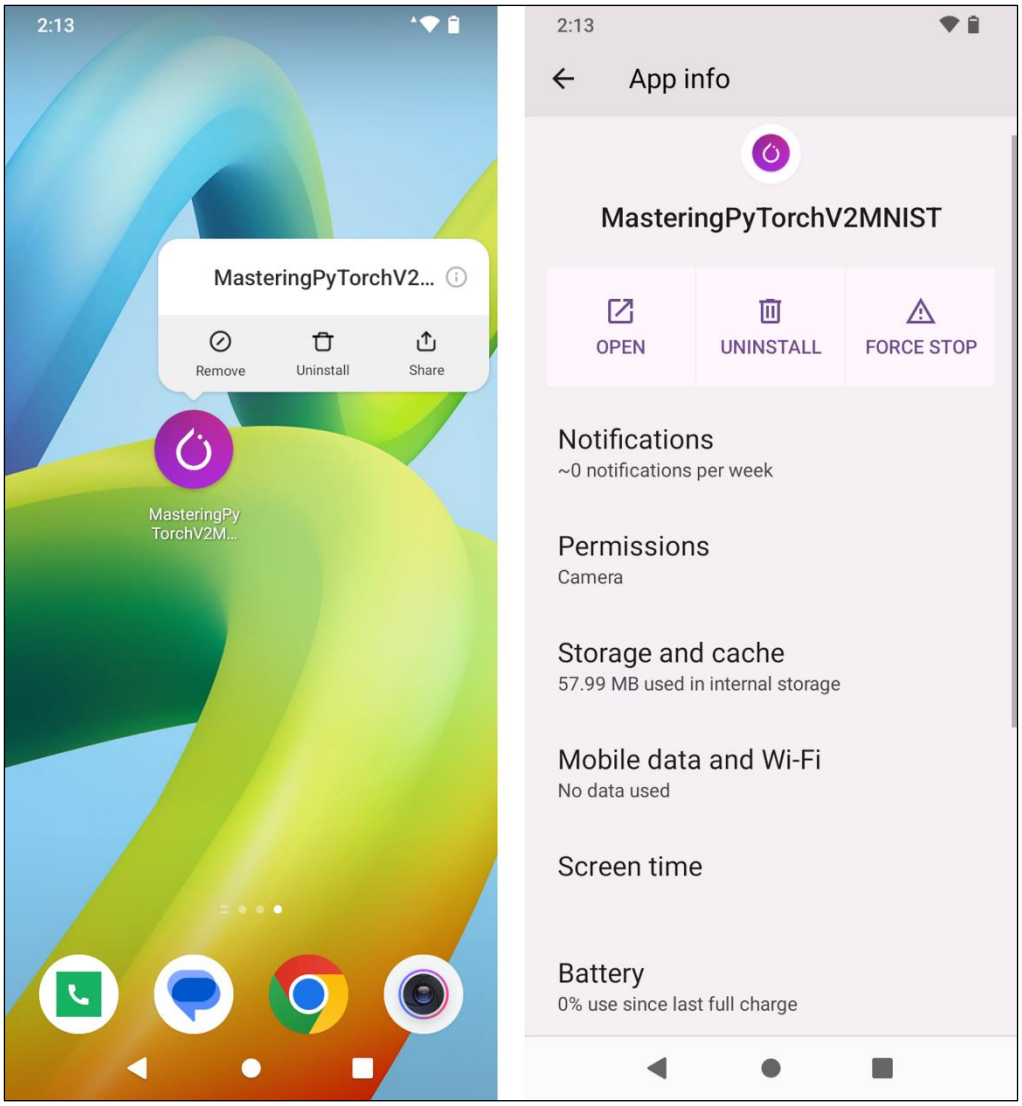


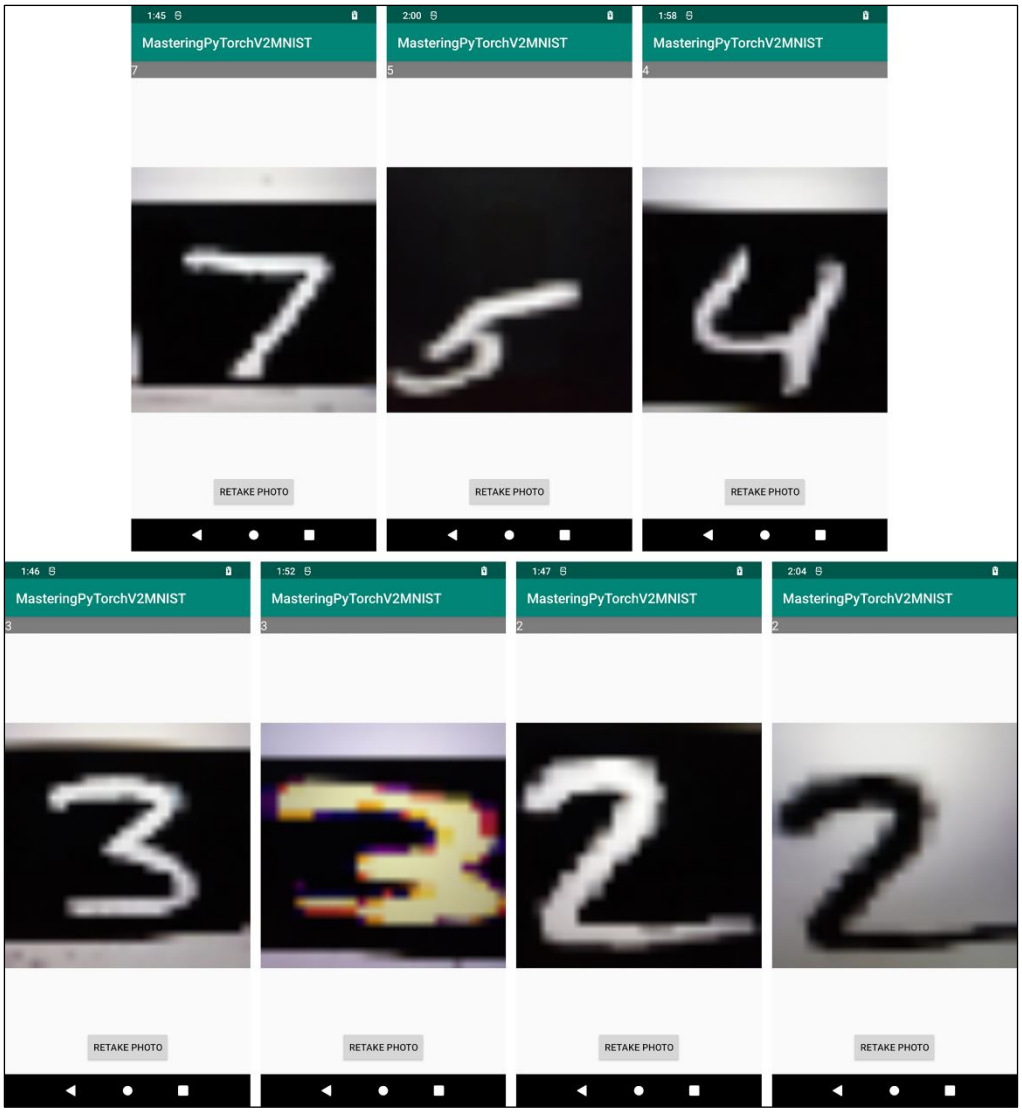

7:09
Camera permission
CAMERA ACCESS FOR THIS APP
Allow only while using the app
Ask every time
Deny
See all apps with this permission

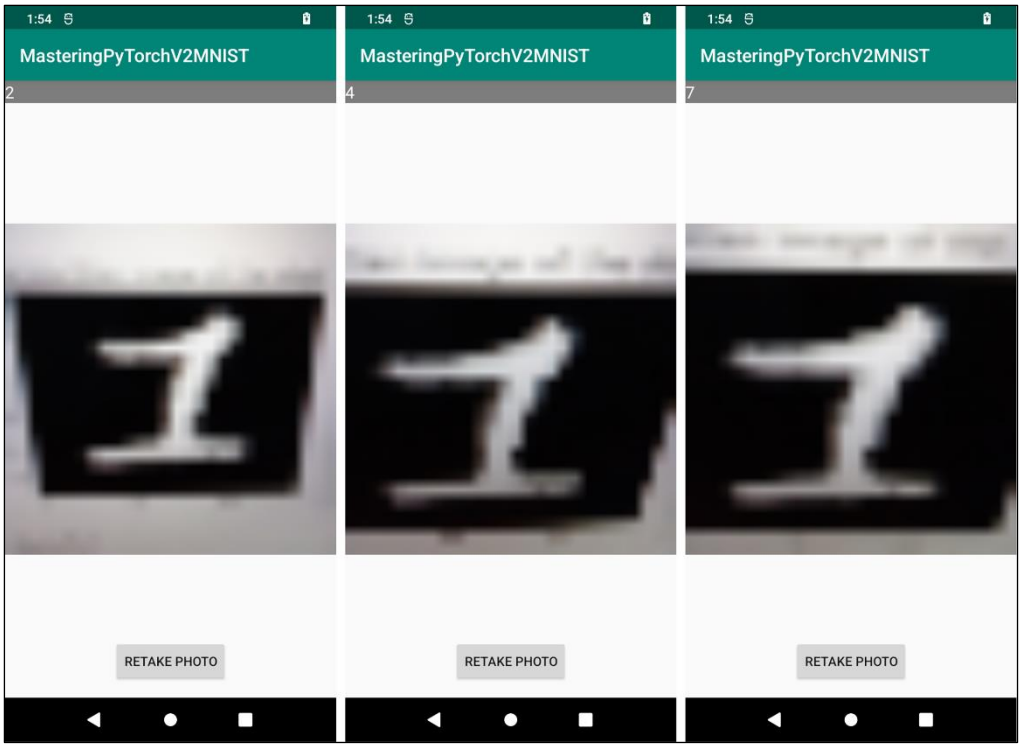

Photo

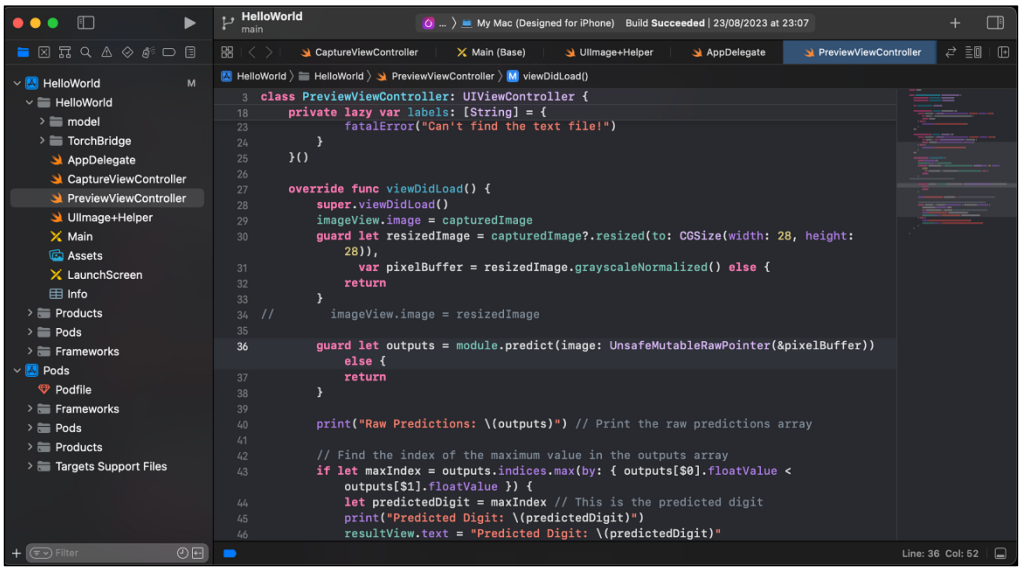

Photo
2:07
MasteringPyTorchV2MNIST
8
RETAKE PHOTO

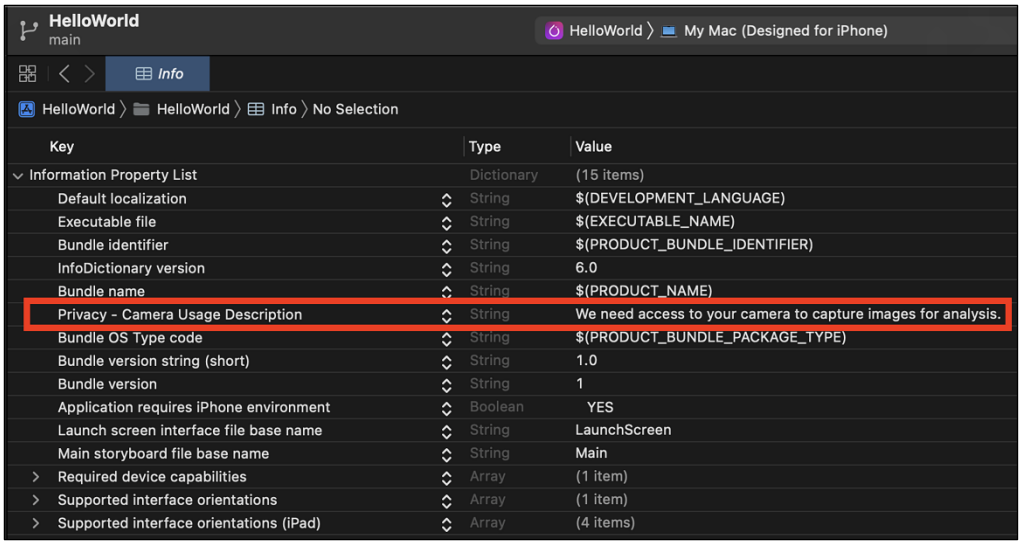

2:14
About phone
Basic info
Device name
Redmi A1
Phone number (sim slot 1)
Unknown
Phone number (sim slot 2)
Unknown
Owner
Legal & regulatory
Certification
Legal information
Device details
SIM status (sim slot 1)
Not available
SIM status (sim slot 2)
Not available
Model
220733SI
IMEI (sim slot 1)
IMEI (sim slot 2)
Android version
12 (Go edition)
Device identifiers
IP address
Wi-Fi MAC address
To view, choose saved network
Device Wi-Fi MAC address
Bluetooth address
Uptime
1108:05:02
Build number
12 V13.0.4.0.SGMINXM

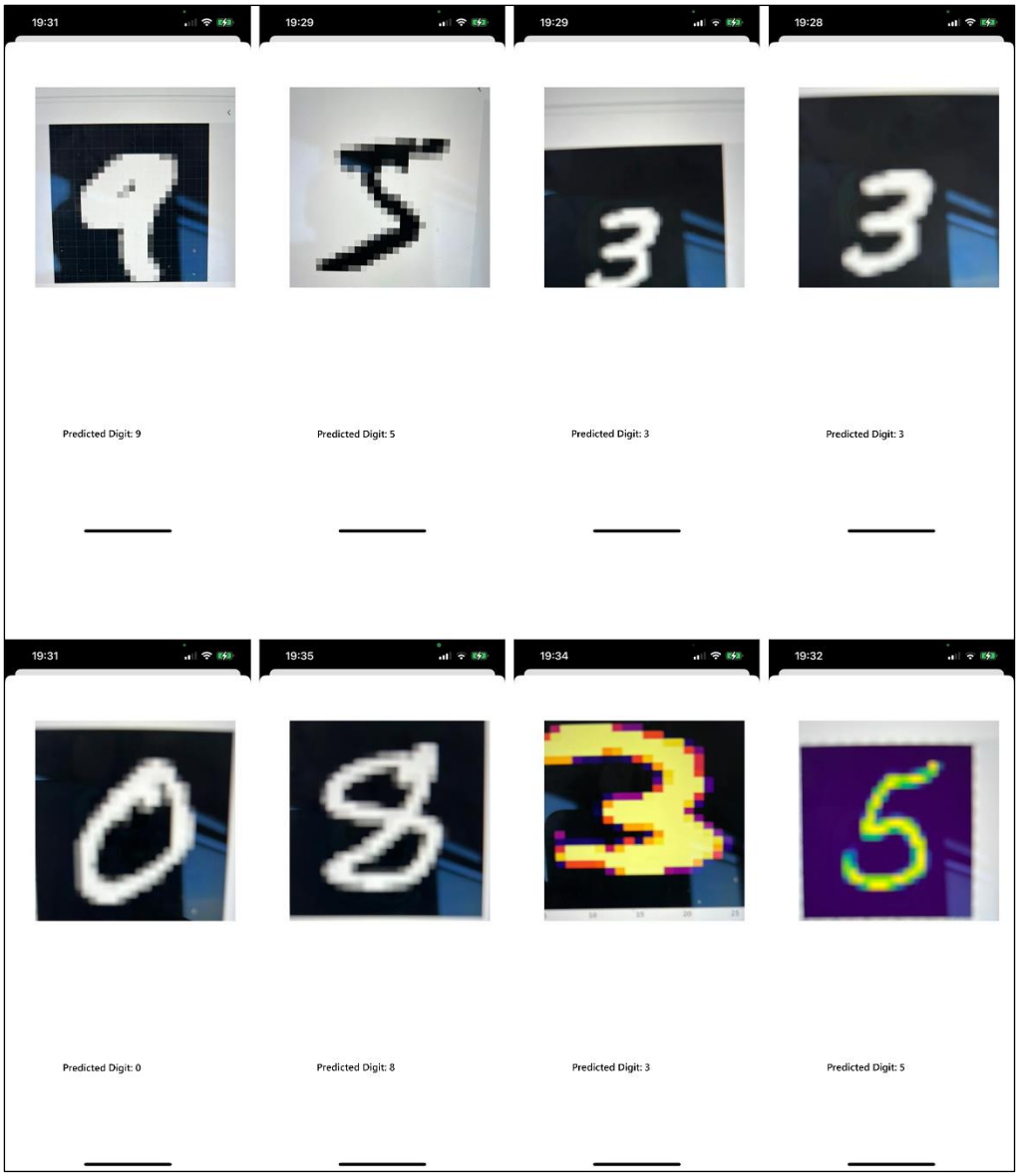

2:13
MasteringPyTorchV2...
Remove
Uninstall
Share
MasteringPy TorchV2M...
2:13
App info
MasteringPyTorchV2MNIST
OPEN
UNINSTALL
FORCE STOP
Notifications
~0 notifications per week
Permissions
Camera
Storage and cache
57.99 MB used in internal storage
Mobile data and Wi-Fi
No data used
Screen time
Battery
0% use since last full charge

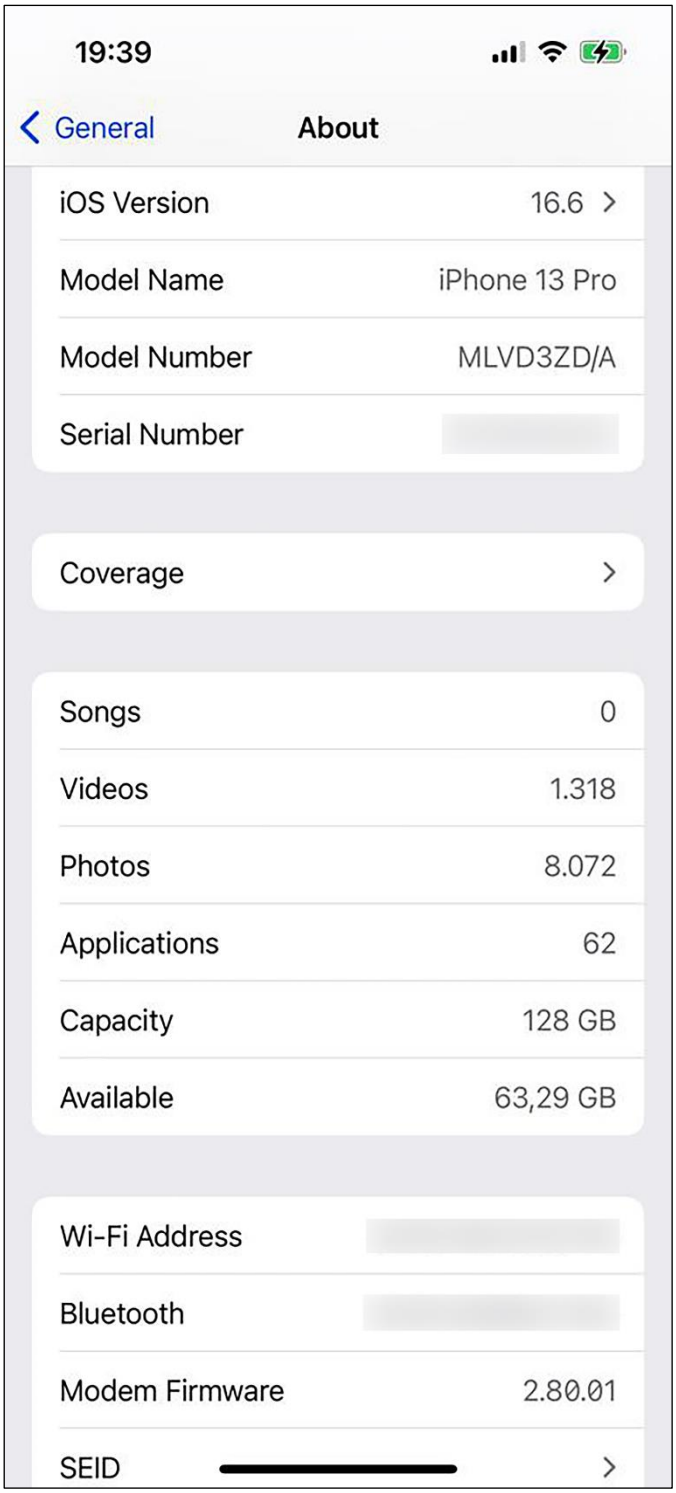

1:45
MasteringPyTorchV2MNIST
7
RETAKE PHOTO
2:00
MasteringPyTorchV2MNIST
5
RETAKE PHOTO
1:58
MasteringPyTorchV2MNIST
4
RETAKE PHOTO
1:46
MasteringPyTorchV2MNIST
3
RETAKE PHOTO
1:52
MasteringPyTorchV2MNIST
3
RETAKE PHOTO
1:47
MasteringPyTorchV2MNIST
2
RETAKE PHOTO
2:04
MasteringPyTorchV2MNIST
2
RETAKE PHOTO


1:54
MasteringPyTorchV2MNIST
2
RETAKE PHOTO
1:54
MasteringPyTorchV2MNIST
4
RETAKE PHOTO
1:54
MasteringPyTorchV2MNIST
7
RETAKE PHOTO

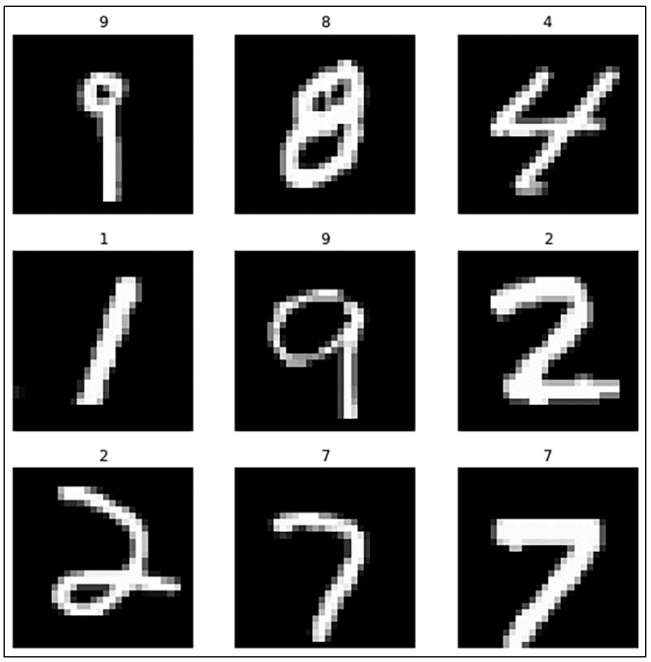

HelloWorld
main
My Mac (Designed for iPhone)
Build Succeeded | 23/08/2023 at 23:07
CaptureViewController
Main (Base)
UIImage+Helper
AppDelegate
PreviewViewController
HelloWorld
model
TorchBridge
AppDelegate
CaptureViewController
PreviewViewController
UIImage+Helper
Main
Assets
LaunchScreen
Info
Products
Pods
Frameworks
Podfile
Targets Support Files
class PreviewViewController: UIViewController {
private lazy var labels: [String] = {
fatalError("Can't find the text file!")
override func viewDidLoad() {
super.viewDidLoad()
imageView.image = capturedImage
guard let resizedImage = capturedImage?.resized(to: CGSize(width: 28, height: 28)),
var pixelBuffer = resizedImage.grayscaleNormalized() else {
return
// imageView.image = resizedImage
guard let outputs = module.predict(image: UnsafeMutableRawPointer(&pixelBuffer))
else {
return
print("Raw Predictions: \(outputs)") // Print the raw predictions array
// Find the index of the maximum value in the outputs array
if let maxIndex = outputs.indices.max(by: { outputs[$0].floatValue < outputs[$1].floatValue }) {
let predictedDigit = maxIndex // This is the predicted digit
print("Predicted Digit: \(predictedDigit)")
resultView.text = "Predicted Digit: \(predictedDigit)"
Line: 36 Col: 52

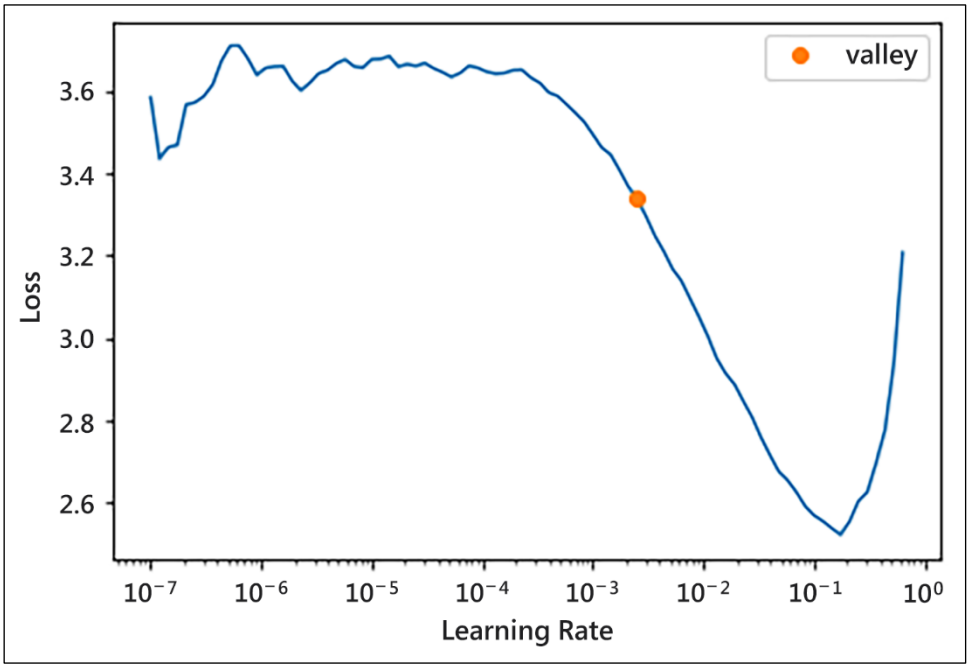

HelloWorld
main
HelloWorld 〉 My Mac (Designed for iPhone)
Info
HelloWorld 〉 HelloWorld 〉 Info 〉 No Selection
Key
Type
Value
Information Property List
Dictionary
(15 items)
Default localization
String
$(DEVELOPMENT_LANGUAGE)
Executable file
String
$(EXECUTABLE_NAME)
Bundle identifier
String
$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version
String
6.0
Bundle name
String
$(PRODUCT_NAME)
Privacy - Camera Usage Description
String
We need access to your camera to capture images for analysis.
Bundle OS Type code
String
$(PRODUCT_BUNDLE_PACKAGE_TYPE)
Bundle version string (short)
String
1.0
Bundle version
String
1
Application requires iPhone environment
Boolean
YES
Launch screen interface file base name
String
LaunchScreen
Main storyboard file base name
String
Main
Required device capabilities
Array
(1 item)
Supported interface orientations
Array
(1 item)
Supported interface orientations (iPad)
Array
(4 items)

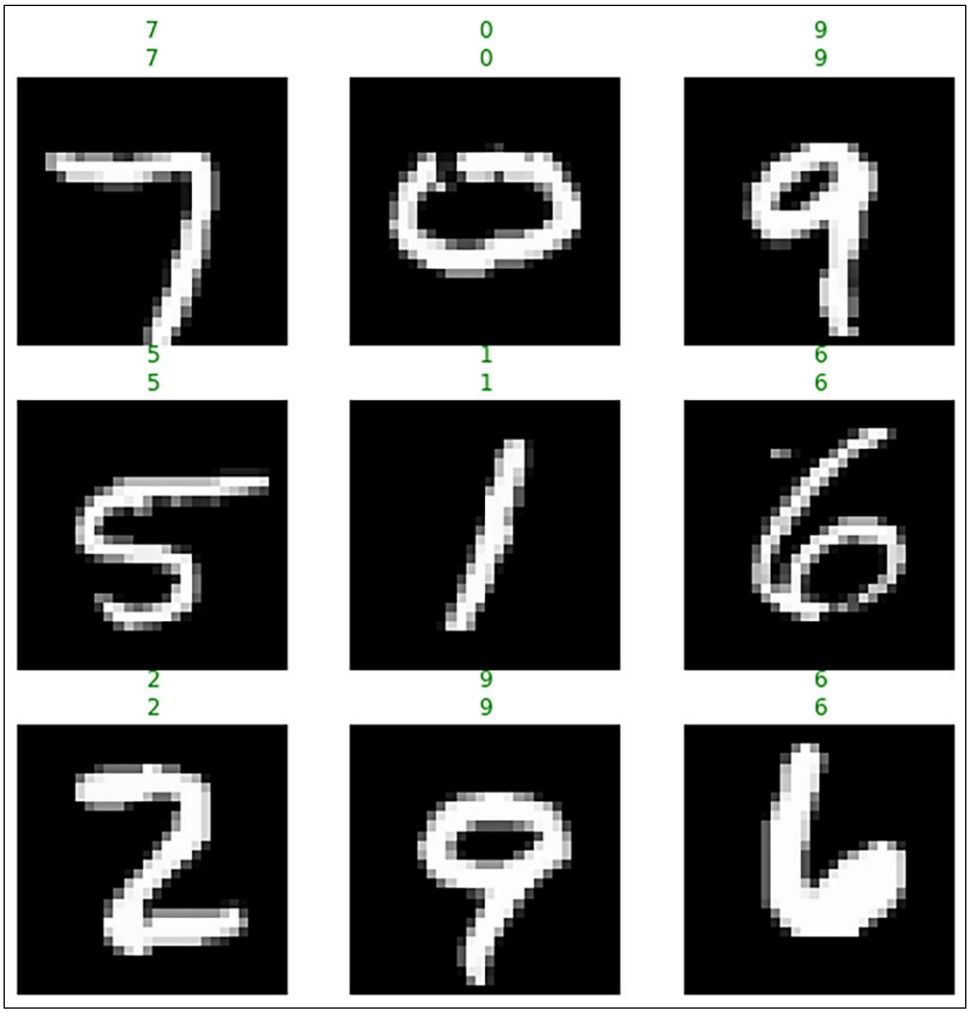

19:31
19:29
19:29
19:28
Predicted Digit: 9
Predicted Digit: 5
Predicted Digit: 3
Predicted Digit: 3
19:31
19:35
19:34
19:32
Predicted Digit: 0
Predicted Digit: 8
Predicted Digit: 3
Predicted Digit: 5

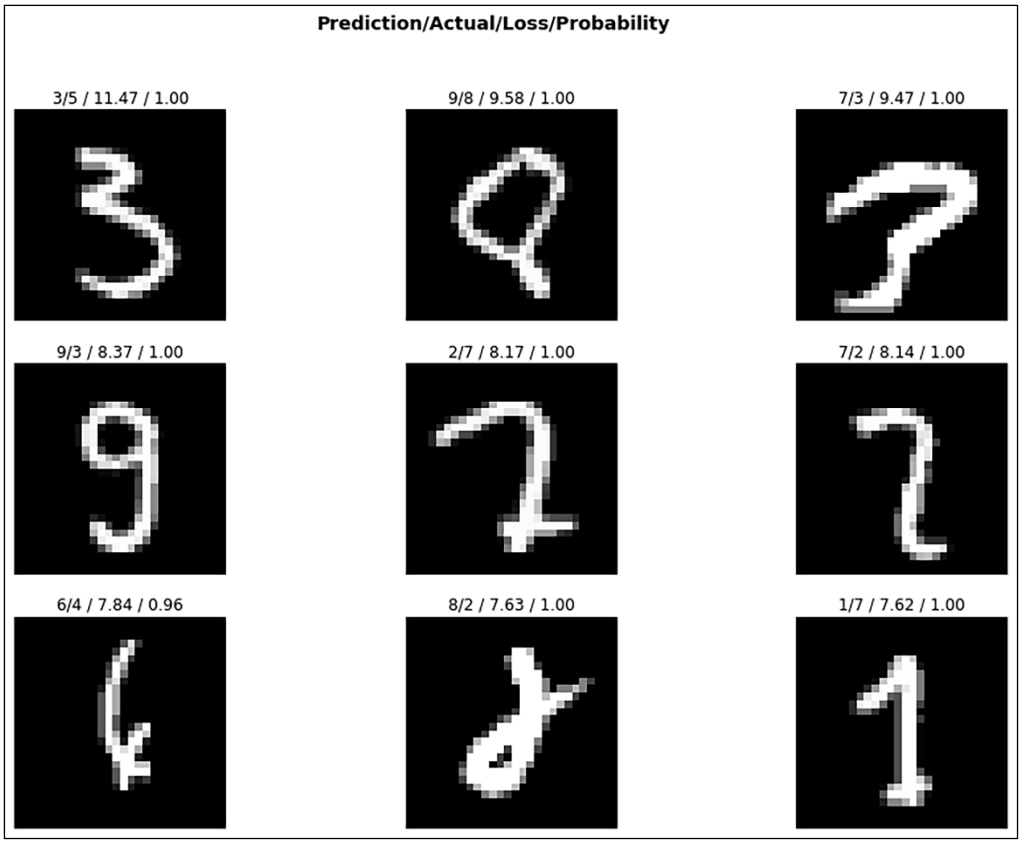

19:39
General
About
iOS Version
16.6
Model Name
iPhone 13 Pro
Model Number
MLVD3ZD/A
Serial Number
Coverage
Songs
0
Videos
1.318
Photos
8.072
Applications
62
Capacity
128 GB
Available
63,29 GB
Wi-Fi Address
Bluetooth
Modem Firmware
2.80.01
SEID

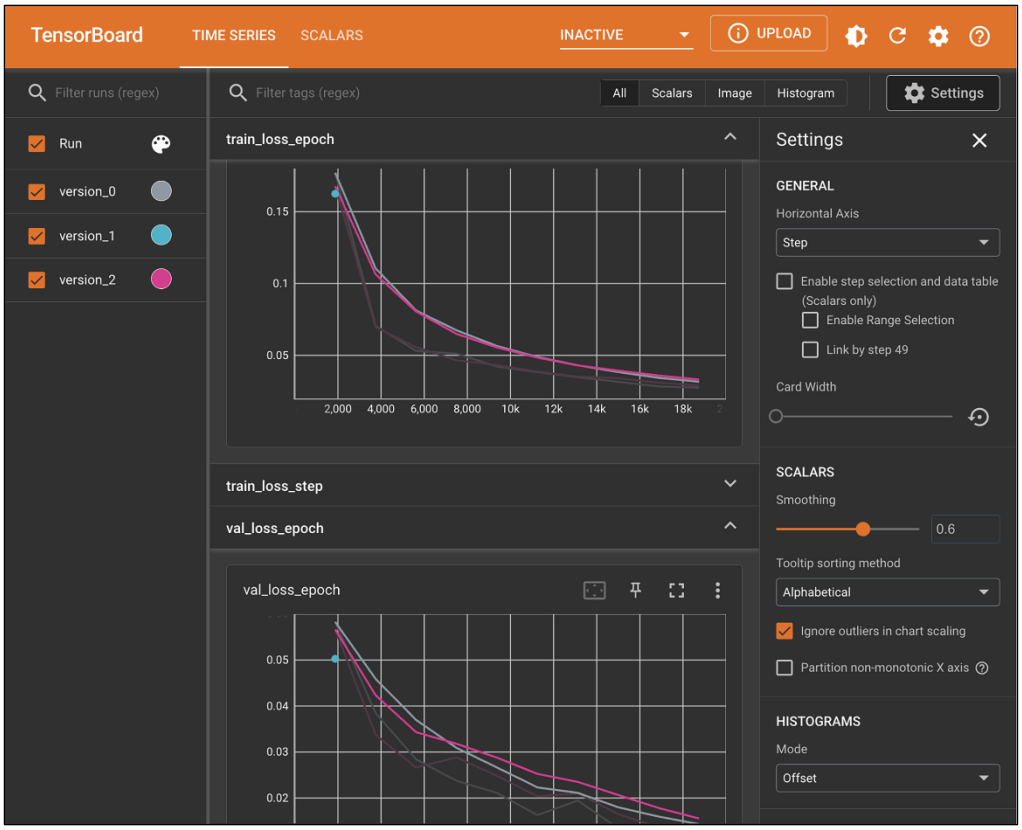
Chapter 15: Rapid Prototyping with PyTorch

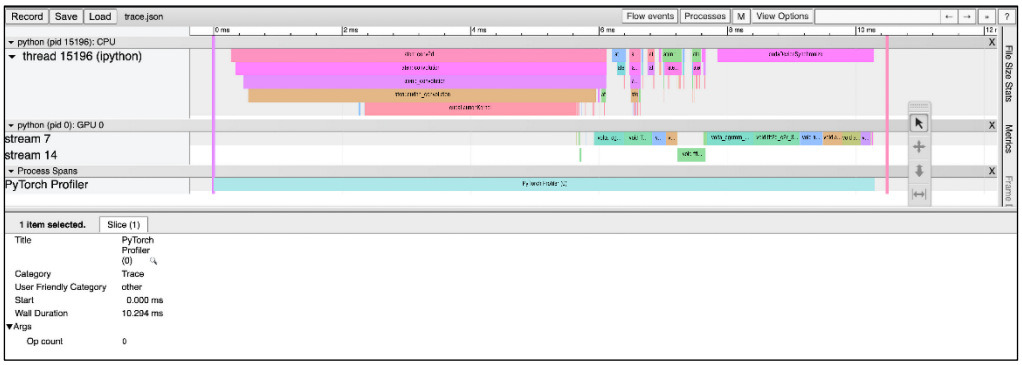




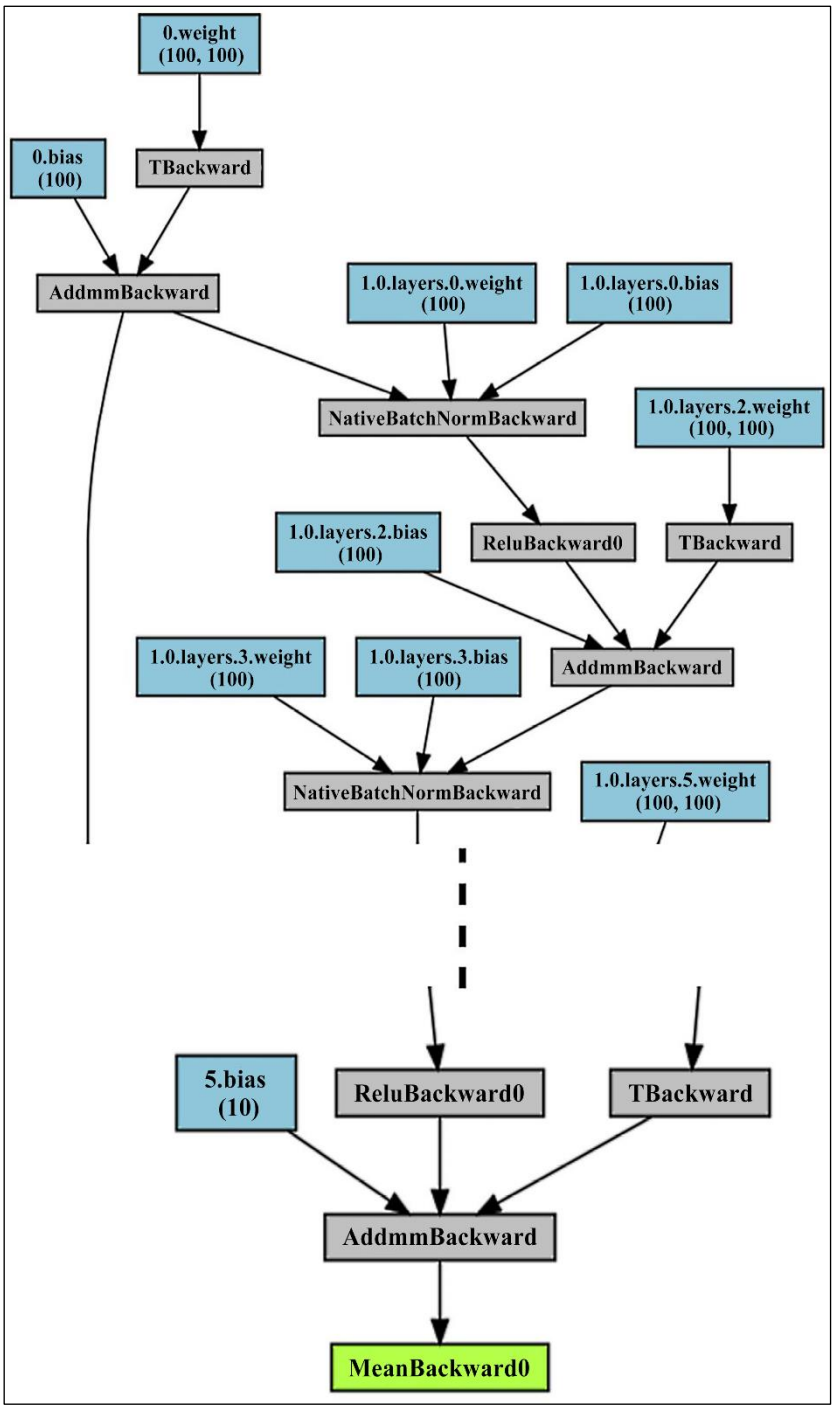

7
7
0
0
9
9
5
5
1
1
6
6
2
2
9
9
6
6


Prediction/Actual/Loss/Probability
3/5 / 11.47 / 1.00
9/8 / 9.58 / 1.00
7/3 / 9.47 / 1.00
9/3 / 8.37 / 1.00
2/7 / 8.17 / 1.00
7/2 / 8.14 / 1.00
6/4 / 7.84 / 0.96
8/2 / 7.63 / 1.00
1/7 / 7.62 / 1.00


TensorBoard
TIME SERIES
SCALARS
INACTIVE
UPLOAD
Filter runs (regex)
Filter tags (regex)
All
Scalars
Image
Histogram
Settings
Run
version_0
version_1
version_2
train_loss_epoch
0.15
0.1
0.05
2,000
4,000
6,000
8,000
10k
12k
14k
16k
18k
train_loss_step
val_loss_epoch
val_loss_epoch
0.05
0.04
0.03
0.02
Settings
GENERAL
Horizontal Axis
Step
Enable step selection and data table (Scalars only)
Enable Range Selection
Link by step 49
Card Width
SCALARS
Smoothing
0.6
Tooltip sorting method
Alphabetical
Ignore outliers in chart scaling
Partition non-monotonic X axis
HISTOGRAMS
Mode
Offset


Record
Save
Load
trace.json
Flow events
Processes
View Options
python (pid 15196): CPU
thread 15196 (ipython)
python (pid 0): GPU 0
stream 7
stream 14
Process Spans
PyTorch Profiler
1 item selected.
Slice (1)
Title
PyTorch Profiler (0)
Category
Trace
User Friendly Category
other
Start
0.000 ms
Wall Duration
10.294 ms
Args
Op count
0

Chapter 16: PyTorch and AutoML

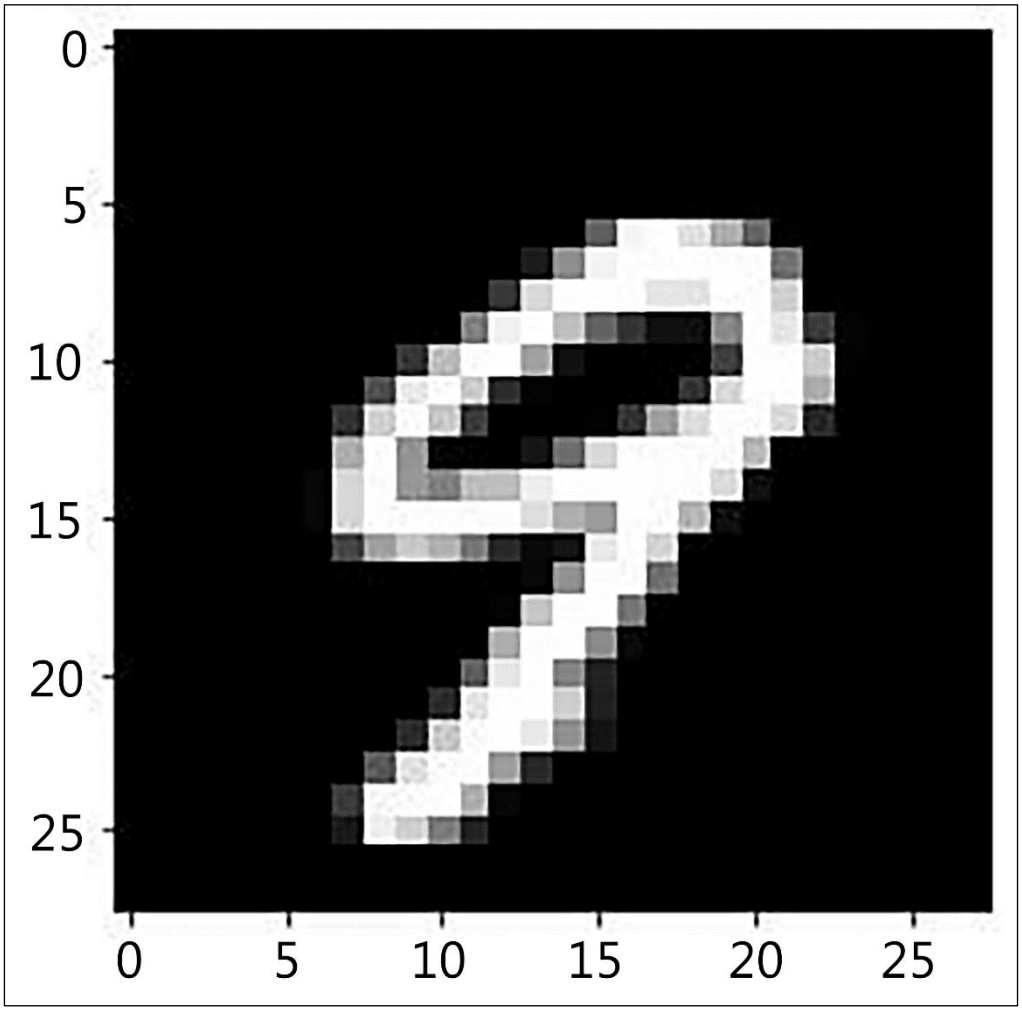


```
epoch: 1 [0/60000 (0%)]  training loss: 2.324445
epoch: 1 [320/60000 (1%)]        training loss: 1.727462
epoch: 1 [640/60000 (1%)]        training loss: 1.428922
epoch: 1 [960/60000 (2%)]        training loss: 0.717944
epoch: 1 [1280/60000 (2%)]       training loss: 0.572199



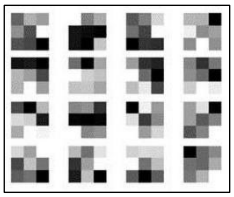
epoch: 19 [58880/60000 (98%)]    training loss: 0.016509
epoch: 19 [59200/60000 (99%)]    training loss: 0.118218
epoch: 19 [59520/60000 (99%)]    training loss: 0.000097
epoch: 19 [59840/60000 (100%)]   training loss: 0.000271

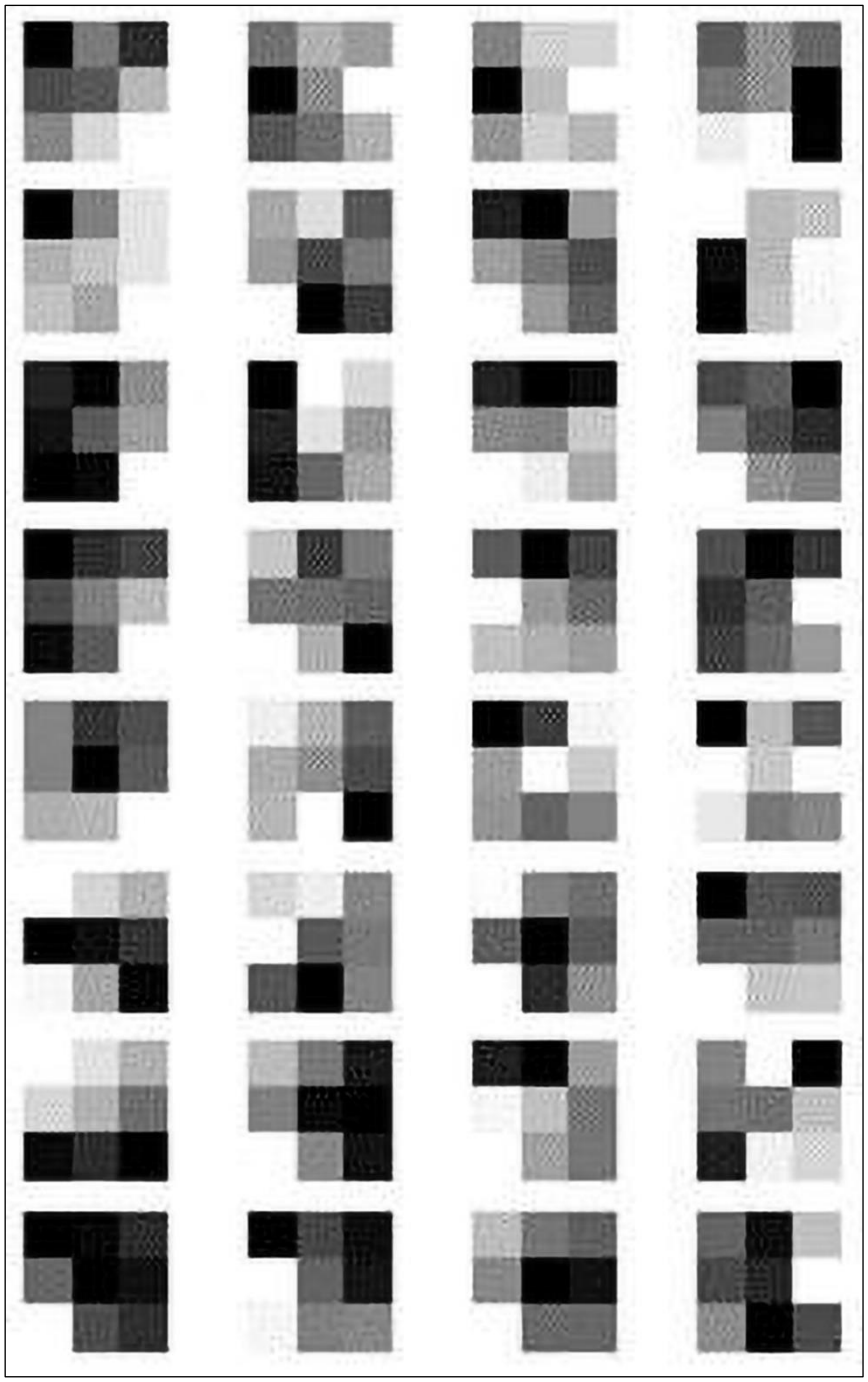
Test dataset: Overall Loss: 0.0387, Overall Accuracy: 9910/10000 (99%)
```



```
Model prediction is : 9
Ground truth is : 9
```

```
[Conv2d(1, 16, kernel_size=(3, 3), stride=(1, 1)),
 Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1)),
 Dropout2d(p=0.1, inplace=False),
 Dropout2d(p=0.25, inplace=False),
 Linear(in_features=4608, out_features=64, bias=True),
 Linear(in_features=64, out_features=10, bias=True)]
```



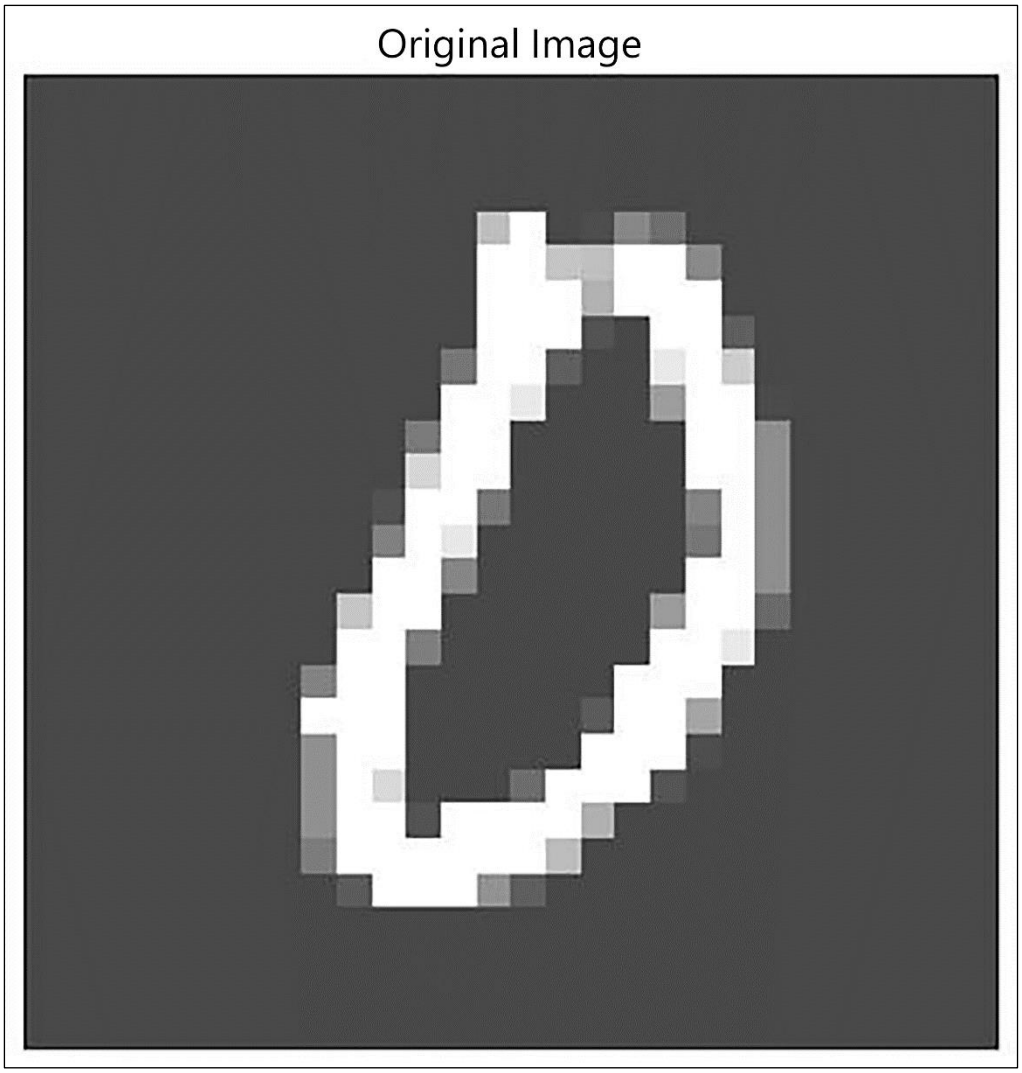


```
torch.Size([16, 26, 26])
```

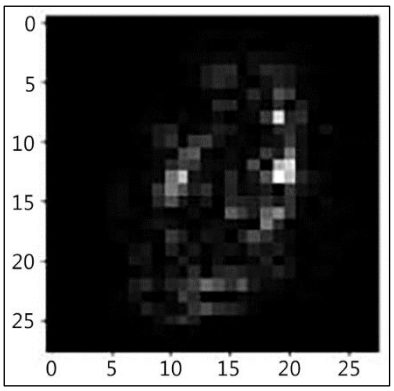


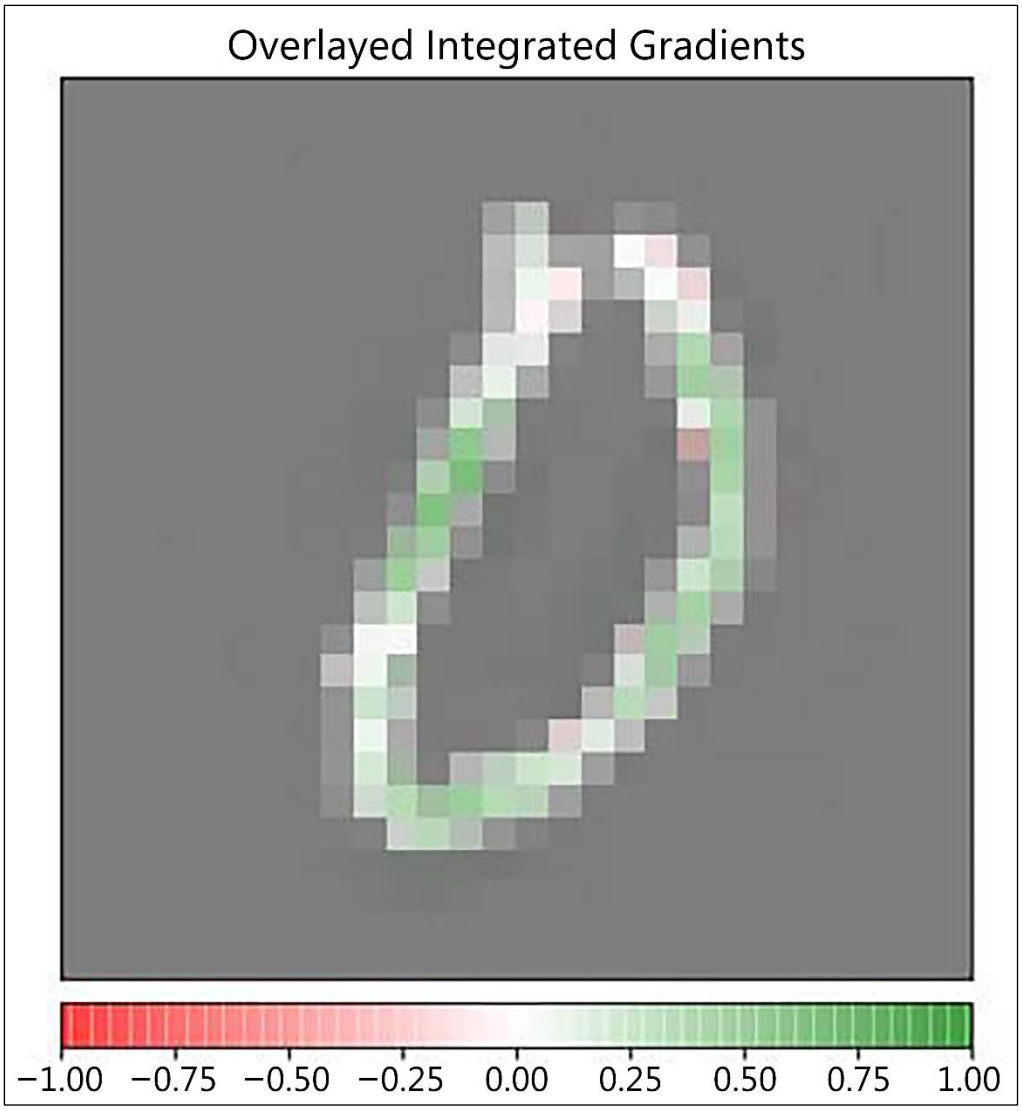
```
torch.Size([32, 24, 24])
```

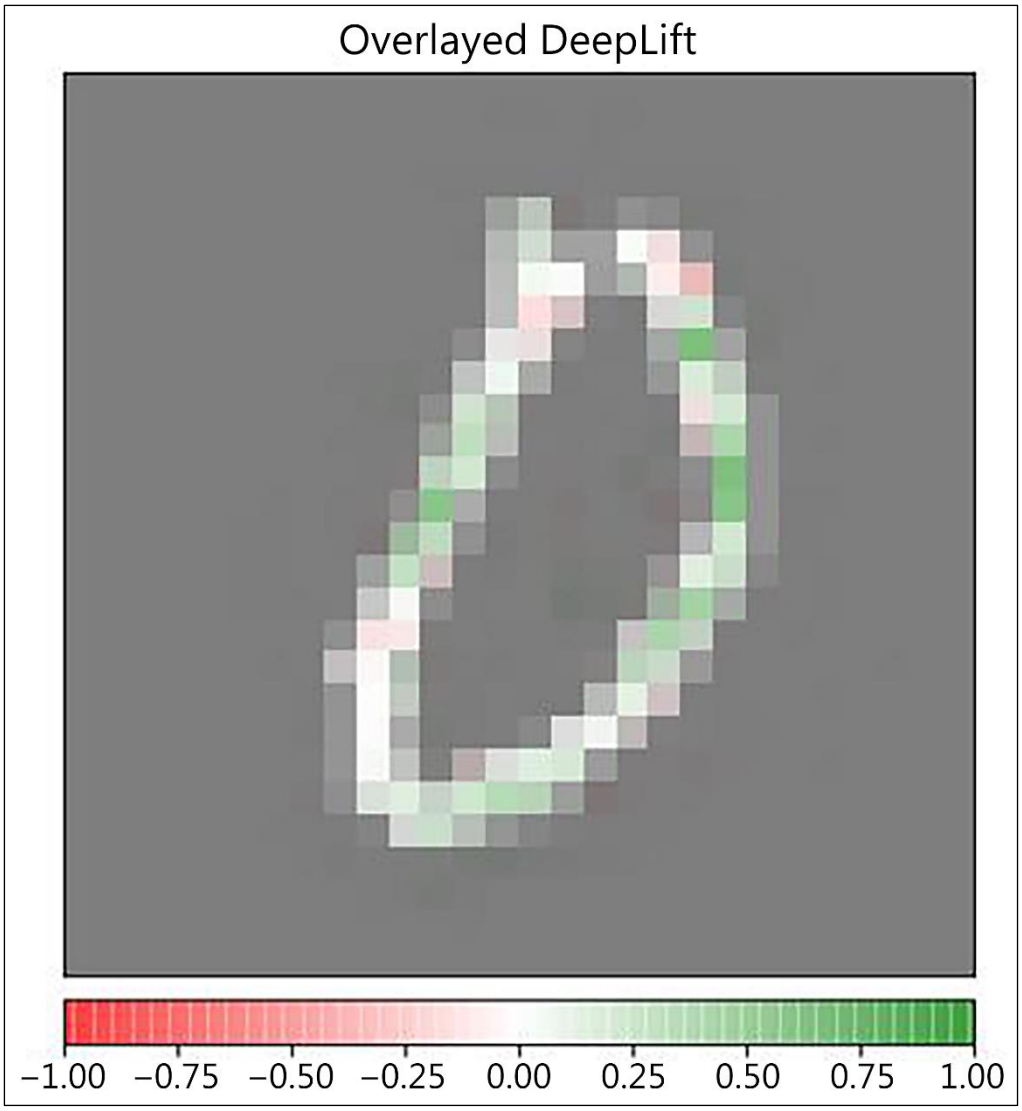



Original Image


Overlayed Gradients
0.0
0.2
0.4
0.6
0.8
1.0


0
5
10
15
20
25
0
5
10
15
20
25


Overlayed Integrated Gradients
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

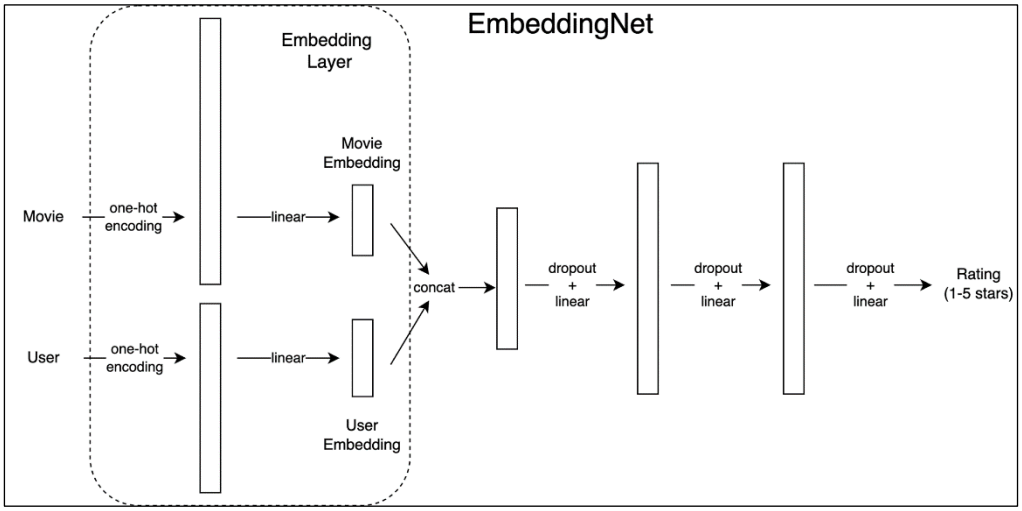

Overlayed DeepLift
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

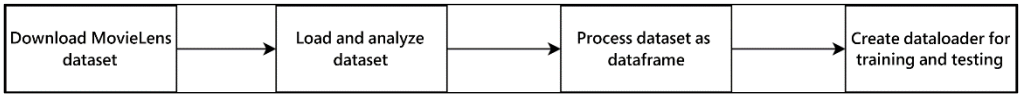
Chapter 18: Recommendation Systems with PyTorch

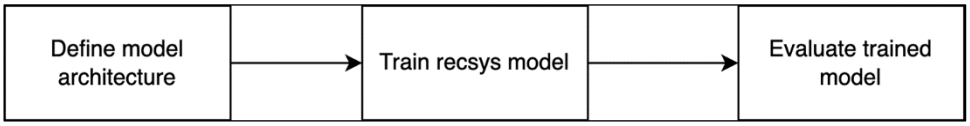
Popular recommendation systems

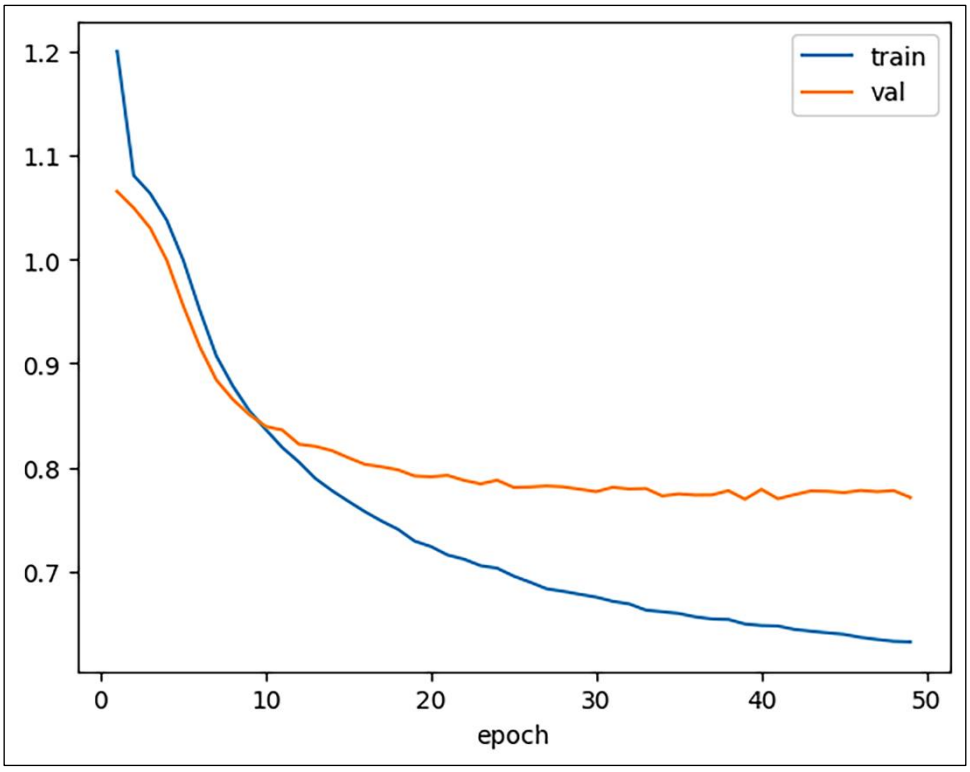


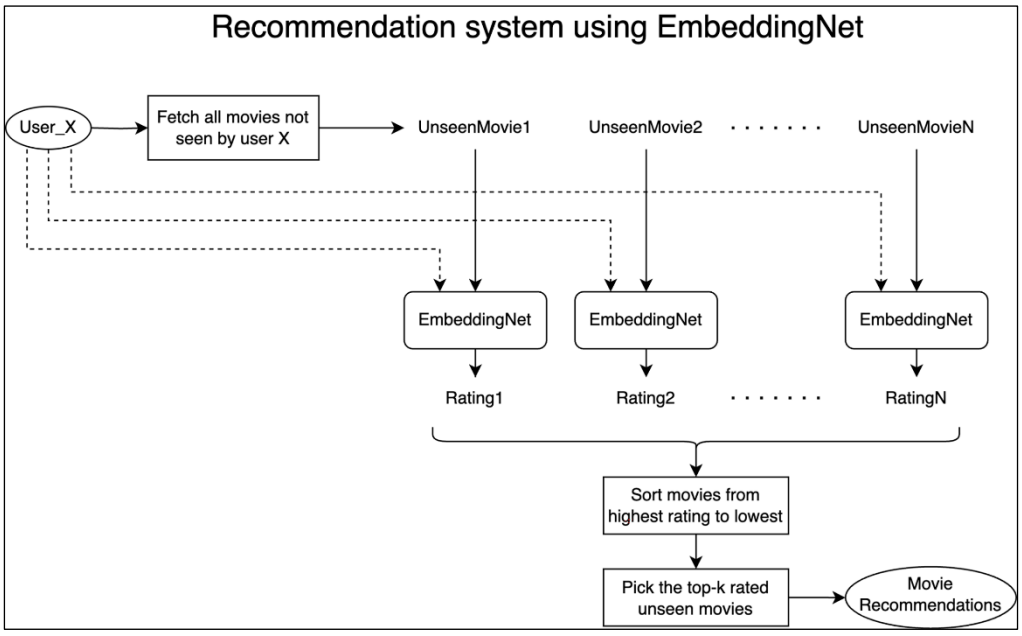
	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6	Movie 7	Movie 8	New Movie
User 1	☆☆☆☆☆	?	?	☆☆	☆☆☆☆	☆☆☆☆	☆☆☆☆	☆☆☆☆☆	?
User 2	?	☆☆☆	☆☆☆☆	☆	?	☆☆☆	?	?	?
User 3	☆☆☆☆	?	☆☆☆	?	☆☆☆☆	?	☆☆☆☆	?	?
User 4	☆☆☆☆	☆☆☆	?	☆☆	?	?	☆☆☆☆	?	?
User 5	?	☆☆☆☆	☆☆	☆☆☆	?	☆☆☆	☆☆	☆☆☆☆☆	?
New User	?	?	?	?	?	?	?	?	?



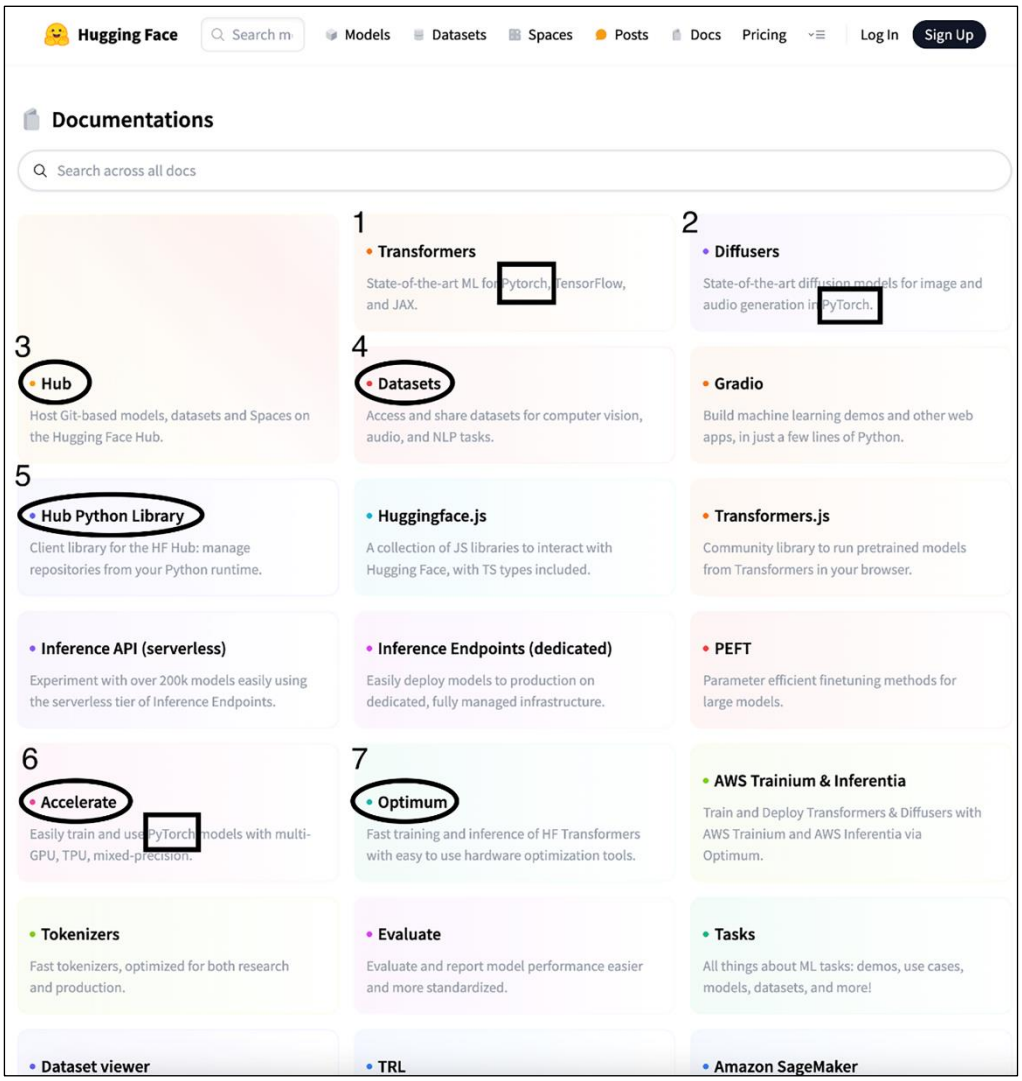


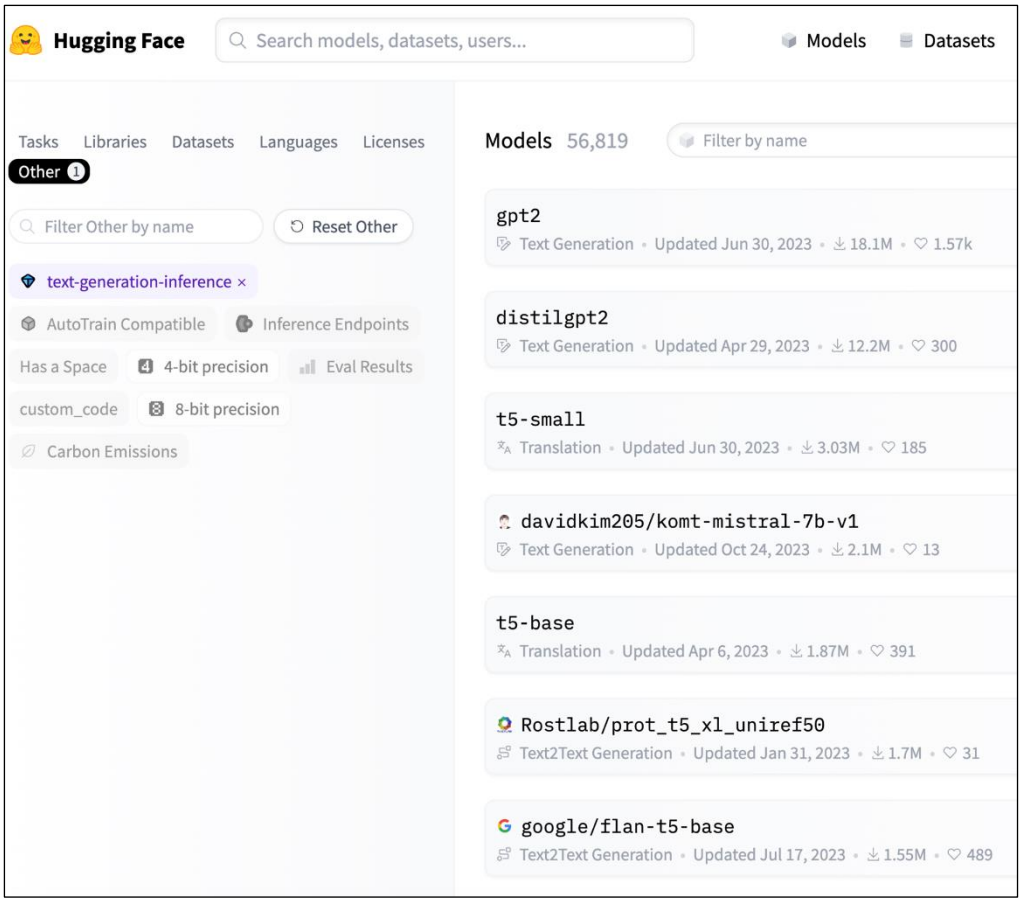



train
val
1.2
1.1
1.0
0.9
0.8
0.7
0
10
20
30
40
50
epoch

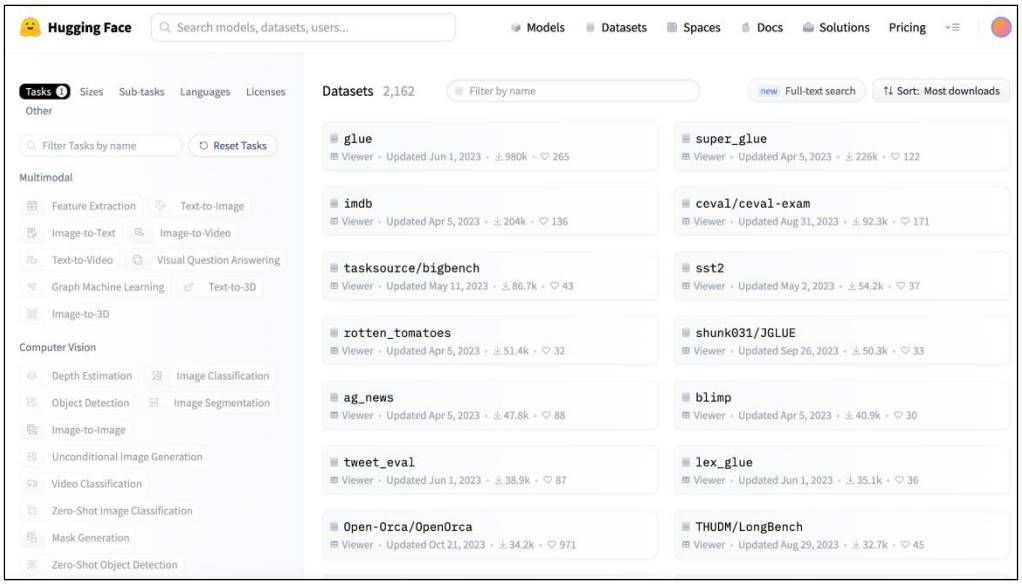

Recommendation system using EmbeddingNet
User_X
Fetch all movies not seen by user X
UnseenMovie1
UnseenMovie2
UnseenMovieN
EmbeddingNet
EmbeddingNet
EmbeddingNet
Rating1
Rating2
RatingN
Sort movies from highest rating to lowest
Pick the top-k rated unseen movies
Movie Recommendations

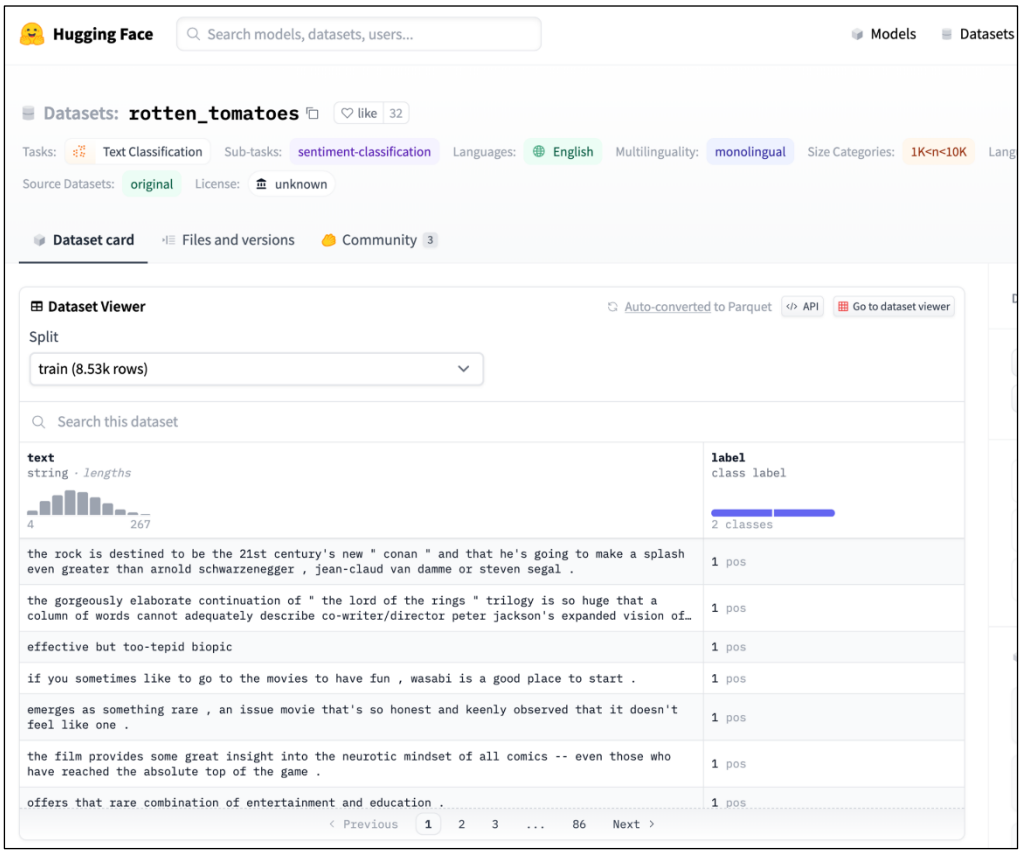
Chapter 19: PyTorch and Hugging Face

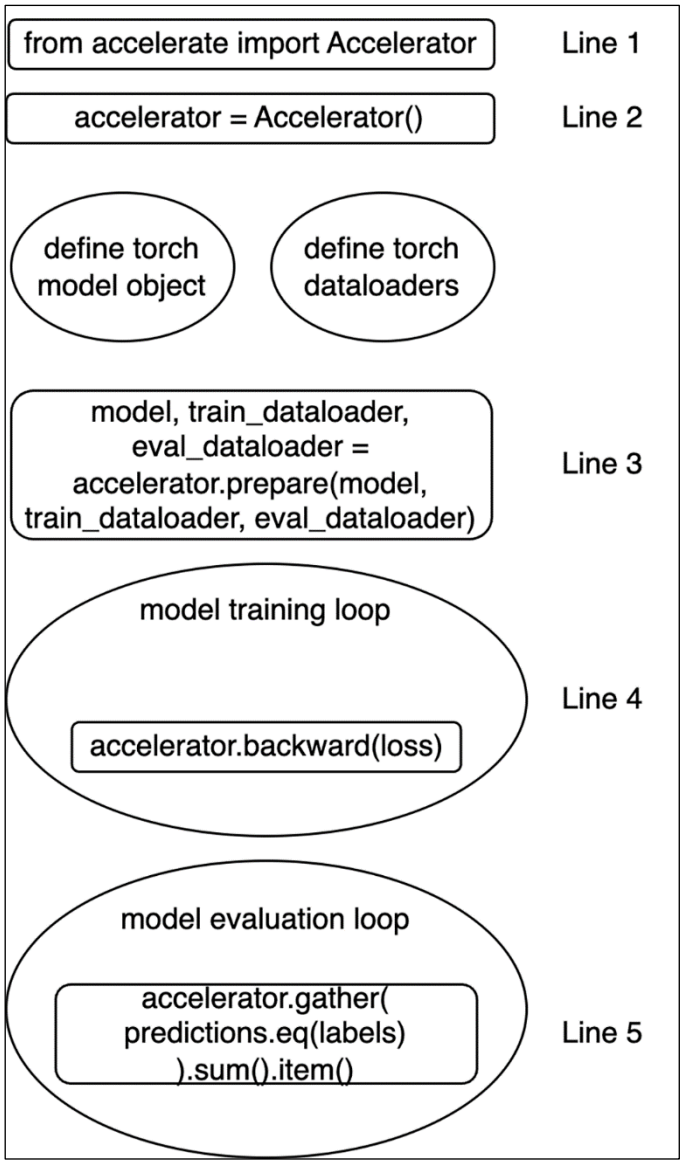


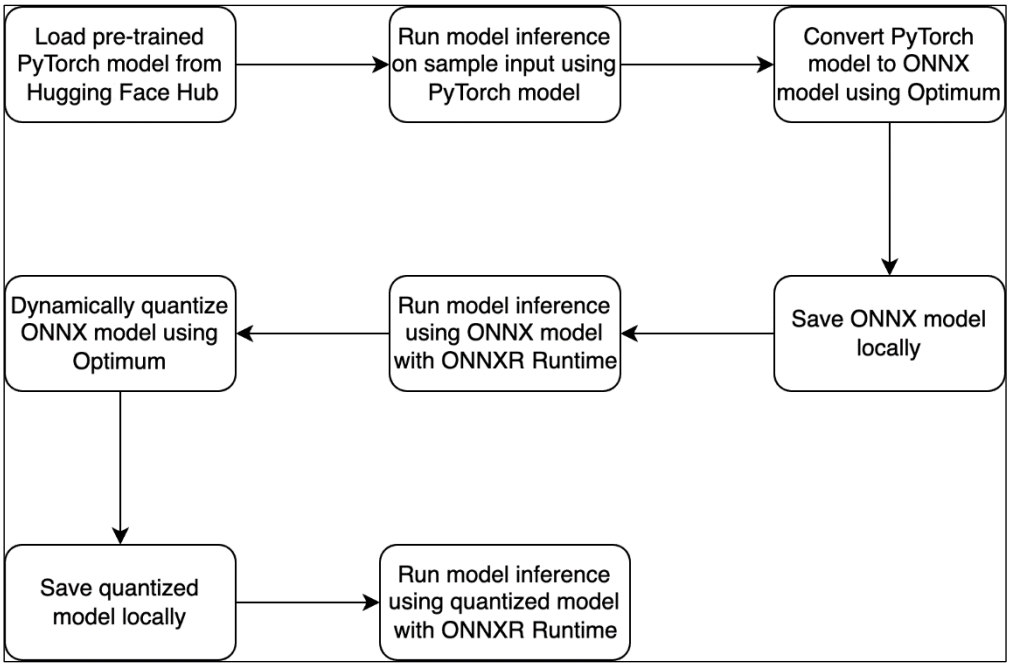

Hugging Face
Search models, datasets, users...
Models
Datasets
Tasks
Libraries
Datasets
Languages
Licenses
Other 1
Filter Other by name
Reset Other
text-generation-inference ×
AutoTrain Compatible
Inference Endpoints
Has a Space
4-bit precision
Eval Results
custom_code
8-bit precision
Carbon Emissions
Models 56,819
Filter by name
gpt2
Text Generation • Updated Jun 30, 2023 • 18.1M • 1.57k
distilgpt2
Text Generation • Updated Apr 29, 2023 • 12.2M • 300
t5-small
Translation • Updated Jun 30, 2023 • 3.03M • 185
davidkim205/komt-mistral-7b-v1
Text Generation • Updated Oct 24, 2023 • 2.1M • 13
t5-base
Translation • Updated Apr 6, 2023 • 1.87M • 391
Rostlab/prot_t5_xl_uniref50
Text2Text Generation • Updated Jan 31, 2023 • 1.7M • 31
google/flan-t5-base
Text2Text Generation • Updated Jul 17, 2023 • 1.55M • 489


Transformers
Search documentation
⌘K
V4.36.1
EN
117,975
RoBERTa
RoBERTa-PreLayerNorm
RoCBert
RoFormer
RWKV
Splinter
SqueezeBERT
SwitchTransformers
T5
T5v1.1
TAPEX
Transformer XL
UL2
UMT5
tokens) and the output sequence is formed as a concatenation of the same sentinel tokens and the real masked tokens. Each sentinel token represents a unique mask token for this sentence and should start with <extra_id_0>, <extra_id_1>, ... up to <extra_id_99>. As a default, 100 sentinel tokens are available in T5Tokenizer.
For instance, the sentence "The cute dog walks in the park" with the masks put on "cute dog" and "the" should be processed as follows:
>>> from transformers import T5Tokenizer, T5ForConditionalGeneration
>>> tokenizer = T5Tokenizer.from_pretrained("t5-small")
>>> model = T5ForConditionalGeneration.from_pretrained("t5-small")
>>> input_ids = tokenizer("The <extra_id_0> walks in <extra_id_1> park", return_tenso
>>> labels = tokenizer("<extra_id_0> cute dog <extra_id_1> the <extra_id_2>", return_
>>> # the forward function automatically creates the correct decoder_input_ids
>>> loss = model(input_ids=input_ids, labels=labels).loss
>>> loss.item()
3.7837


Hugging Face
Search models, datasets, users...
Models
Datasets
Spaces
Docs
Solutions
Pricing
Tasks
Sizes
Sub-tasks
Languages
Licenses
Other
Filter Tasks by name
Reset Tasks
Multimodal
Feature Extraction
Text-to-Image
Image-to-Text
Image-to-Video
Text-to-Video
Visual Question Answering
Graph Machine Learning
Text-to-3D
Image-to-3D
Computer Vision
Depth Estimation
Image Classification
Object Detection
Image Segmentation
Image-to-Image
Unconditional Image Generation
Video Classification
Zero-Shot Image Classification
Mask Generation
Zero-Shot Object Detection
Datasets 2,162
Filter by name
new Full-text search
Sort: Most downloads
glue
Viewer • Updated Jun 1, 2023 • 980k • 265
super_glue
Viewer • Updated Apr 5, 2023 • 226k • 122
imdb
Viewer • Updated Apr 5, 2023 • 204k • 136
ceval/ceval-exam
Viewer • Updated Aug 31, 2023 • 92.3k • 171
tasksource/bigbench
Viewer • Updated May 11, 2023 • 86.7k • 43
sst2
Viewer • Updated May 2, 2023 • 54.2k • 37
rotten_tomatoes
Viewer • Updated Apr 5, 2023 • 51.4k • 32
shunk031/JGLUE
Viewer • Updated Sep 26, 2023 • 50.3k • 33
ag_news
Viewer • Updated Apr 5, 2023 • 47.8k • 88
blimp
Viewer • Updated Apr 5, 2023 • 40.9k • 30
tweet_eval
Viewer • Updated Jun 1, 2023 • 38.9k • 87
lex_glue
Viewer • Updated Jun 1, 2023 • 35.1k • 36
Open-Orca/OpenOrca
Viewer • Updated Oct 21, 2023 • 34.2k • 971
THUDM/LongBench
Viewer • Updated Aug 29, 2023 • 32.7k • 45


Hugging Face
Search models, datasets, users...
Models
Datasets
Datasets: rotten_tomatoes
like 32
Tasks: Text Classification
Sub-tasks: sentiment-classification
Languages: English
Multilinguality: monolingual
Size Categories: 1K<n<10K
Source Datasets: original
License: unknown
Dataset card
Files and versions
Community 3
Dataset Viewer
Auto-converted to Parquet
API
Go to dataset viewer
Split
train (8.53k rows)
Search this dataset
text
string · lengths
4
267
label
class label
2 classes
the rock is destined to be the 21st century's new " conan " and that he's going to make a splash even greater than arnold schwarzenegger , jean-claud van damme or steven segal .
1 pos
the gorgeously elaborate continuation of " the lord of the rings " trilogy is so huge that a column of words cannot adequately describe co-writer/director peter jackson's expanded vision of…
1 pos
effective but too-tepid biopic
1 pos
if you sometimes like to go to the movies to have fun , wasabi is a good place to start .
1 pos
emerges as something rare , an issue movie that's so honest and keenly observed that it doesn't feel like one .
1 pos
the film provides some great insight into the neurotic mindset of all comics -- even those who have reached the absolute top of the game .
1 pos
offers that rare combination of entertainment and education .
1 pos
Previous
1
2
3
...
86
Next


from accelerate import Accelerator
Line 1
accelerator = Accelerator()
Line 2
define torch model object
define torch dataloaders
model, train_dataloader, eval_dataloader = accelerator.prepare(model, train_dataloader, eval_dataloader)
Line 3
model training loop
accelerator.backward(loss)
Line 4
model evaluation loop
accelerator.gather(predictions.eq(labels)).sum().item()
Line 5


Load pre-trained PyTorch model from Hugging Face Hub
Run model inference on sample input using PyTorch model
Convert PyTorch model to ONNX model using Optimum
Save ONNX model locally
Run model inference using ONNX model with ONNXR Runtime
Dynamically quantize ONNX model using Optimum
Save quantized model locally
Run model inference using quantized model with ONNXR Runtime