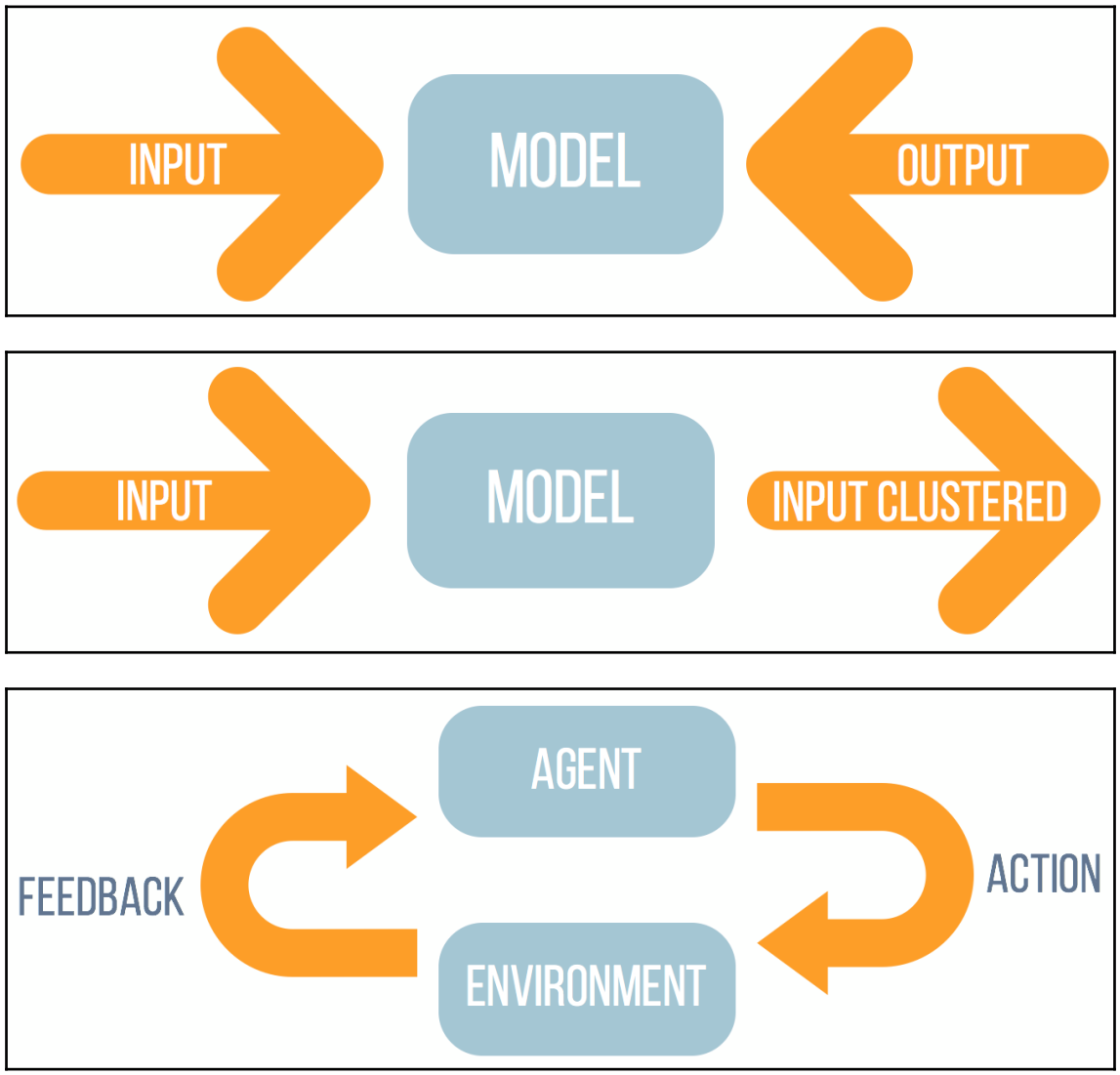
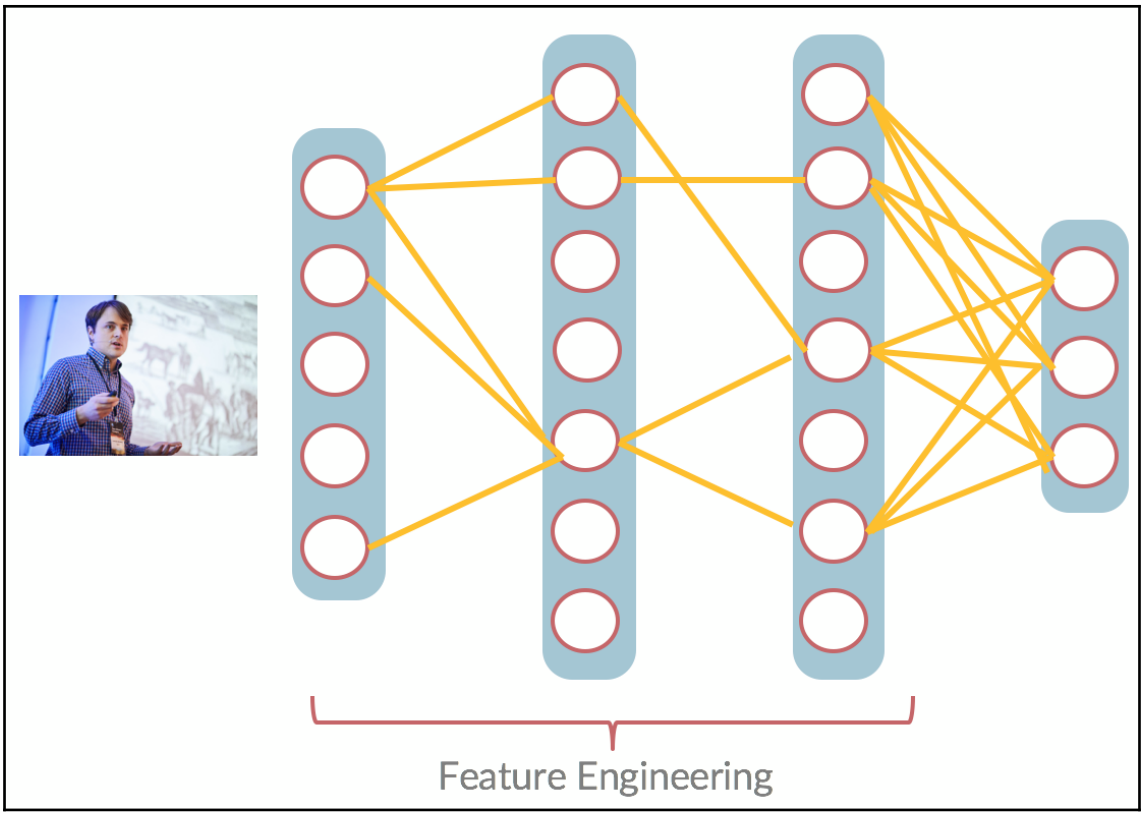
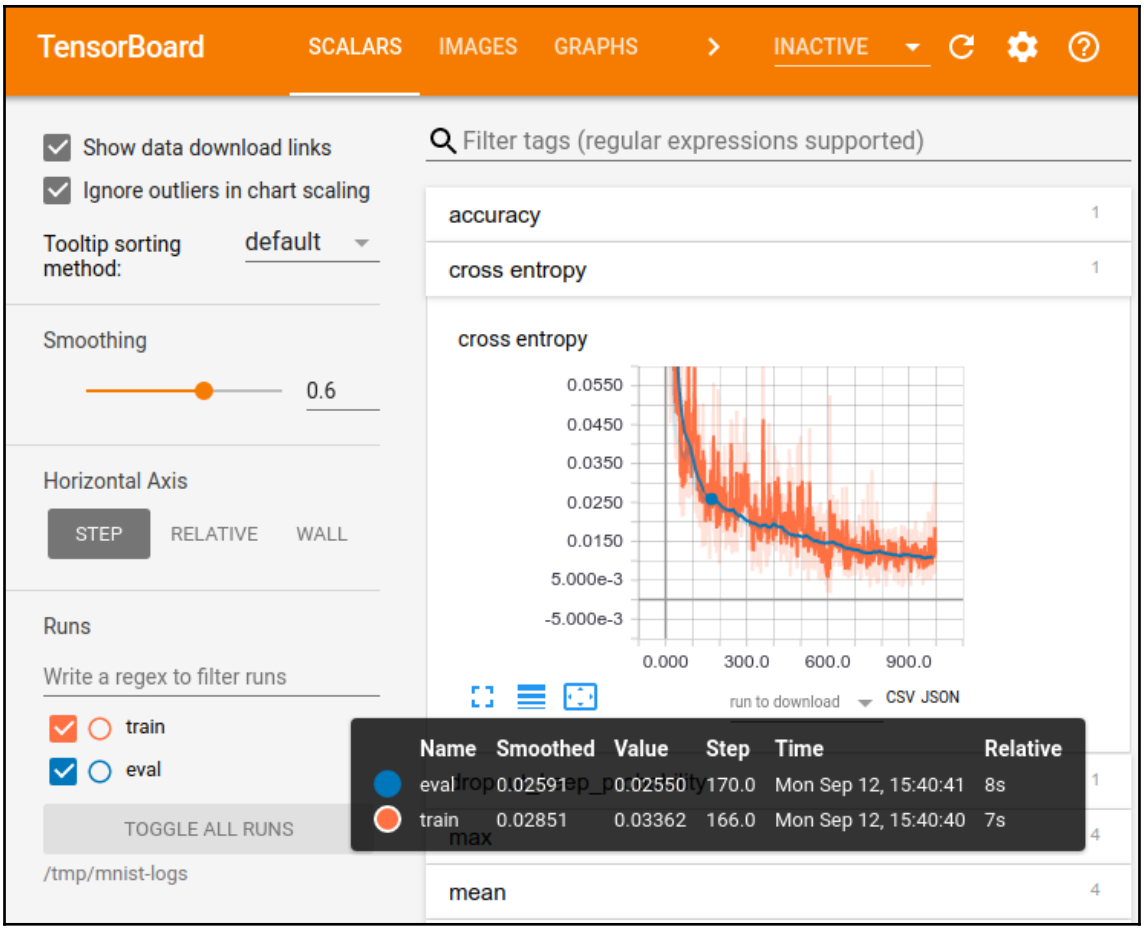
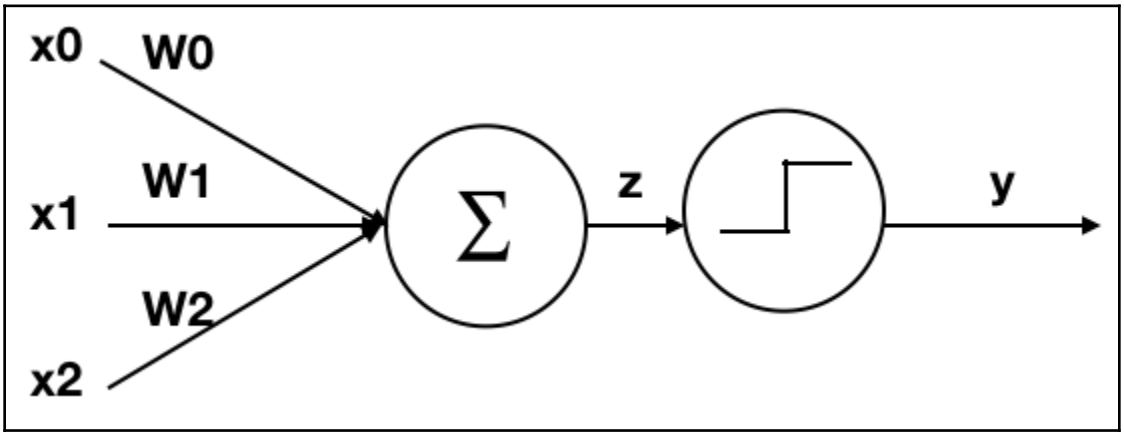
Chapter 1: Getting Started with Supervised Learning

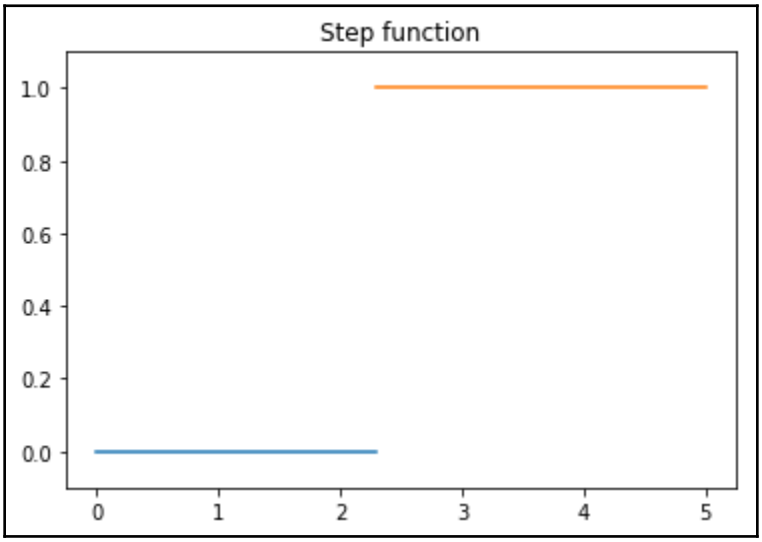


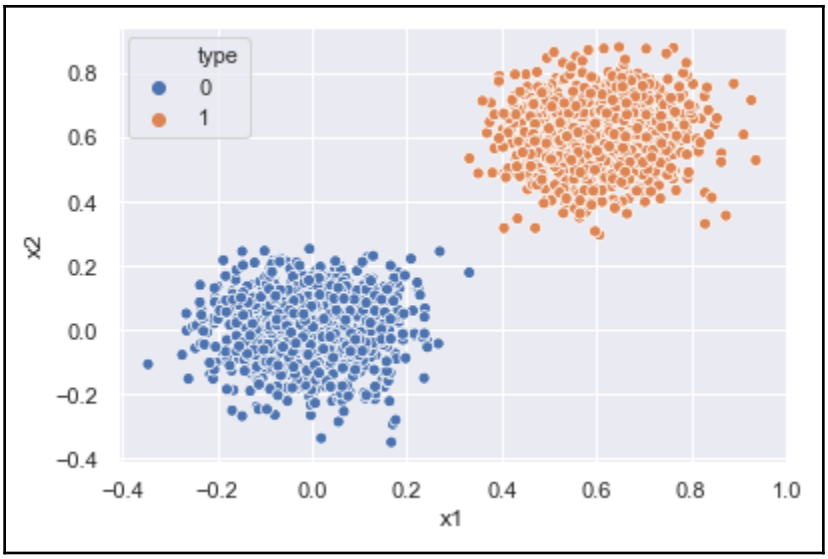

Feature Engineering

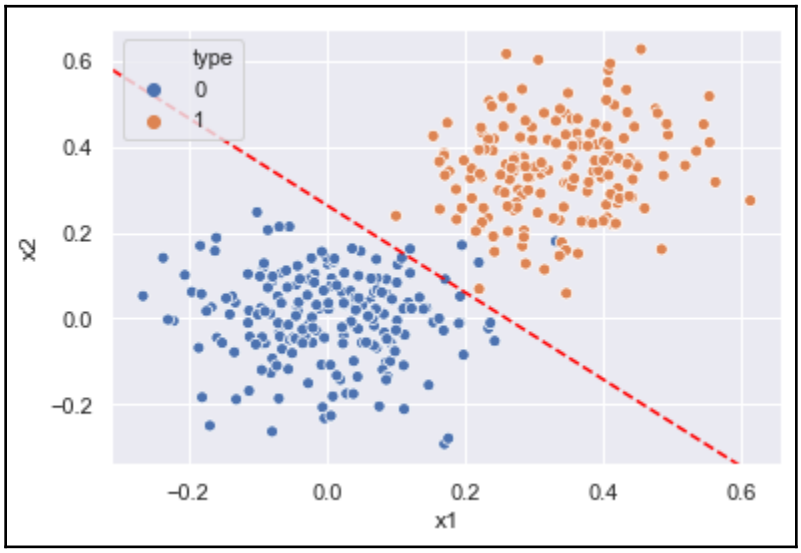

TensorBoard
SCALARS
IMAGES
GRAPHS
INACTIVE
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method:
default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
train
eval
TOGGLE ALL RUNS
/tmp/mnist-logs
Filter tags (regular expressions supported)
accuracy
1
cross entropy
1
cross entropy
0.0550
0.0450
0.0350
0.0250
0.0150
5.000e-3
-5.000e-3
0.000
300.0
600.0
900.0
run to download
CSV JSON
Name Smoothed Value Step Time Relative
eval 0.02591 0.02550 170.0 Mon Sep 12, 15:40:41 8s
train 0.02851 0.03362 166.0 Mon Sep 12, 15:40:40 7s
max
4
mean
4

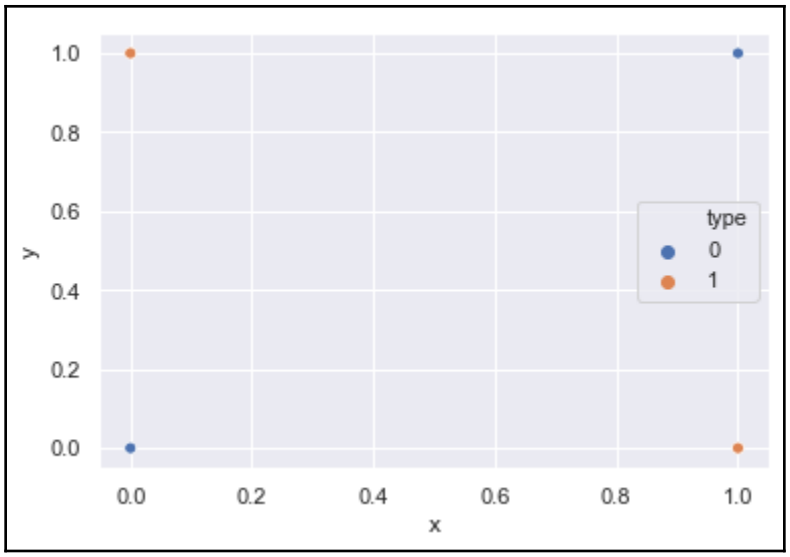
Chapter 2: Neural Network Fundamentals

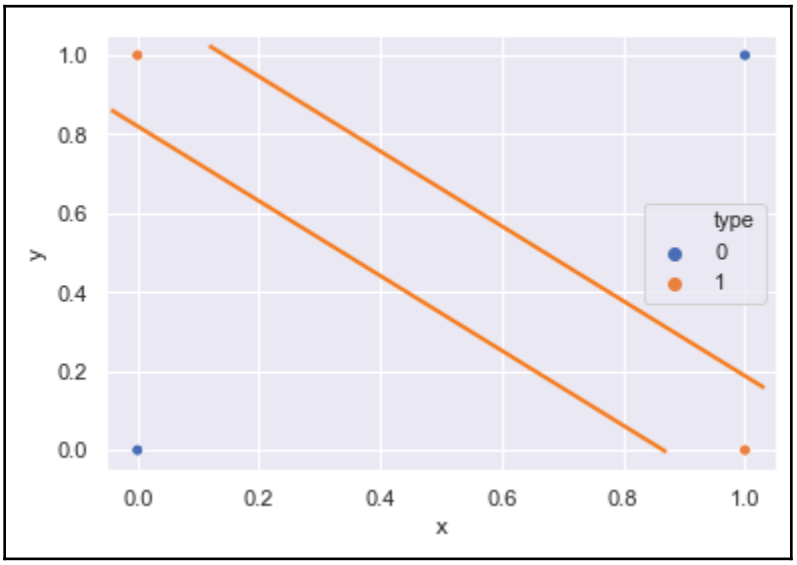


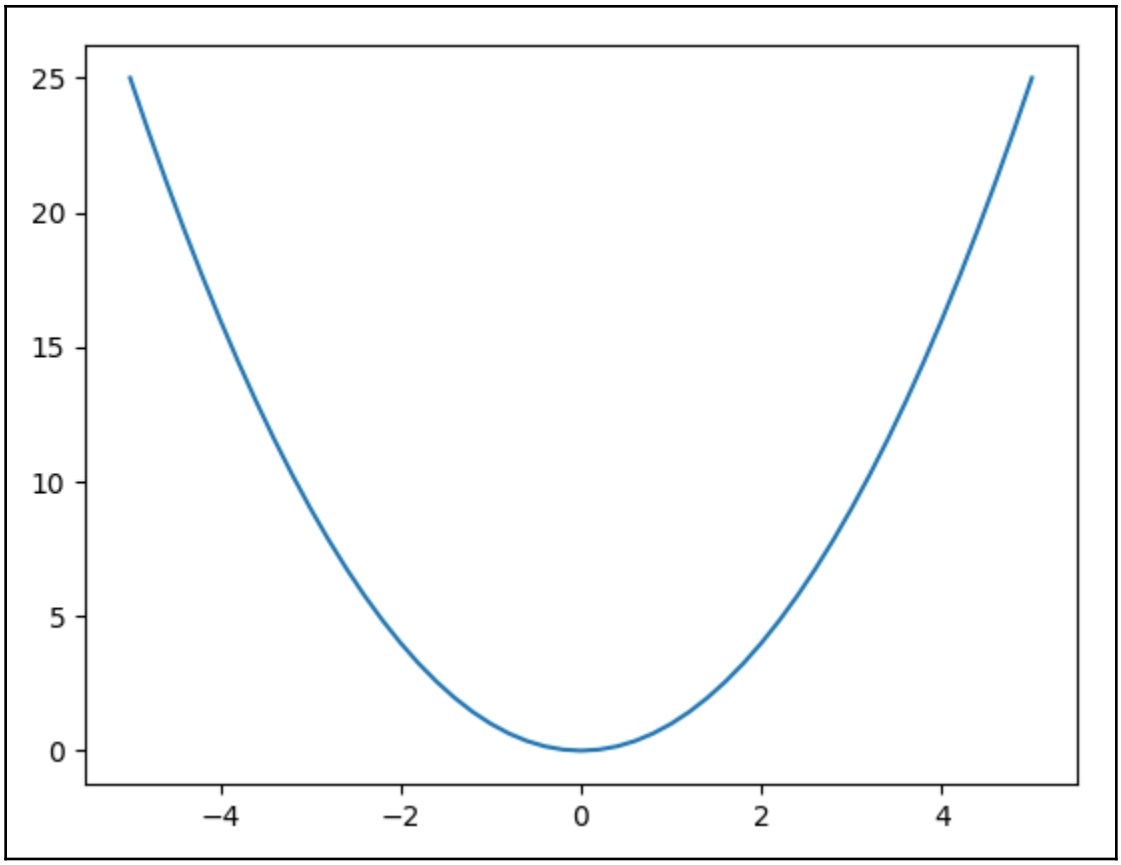


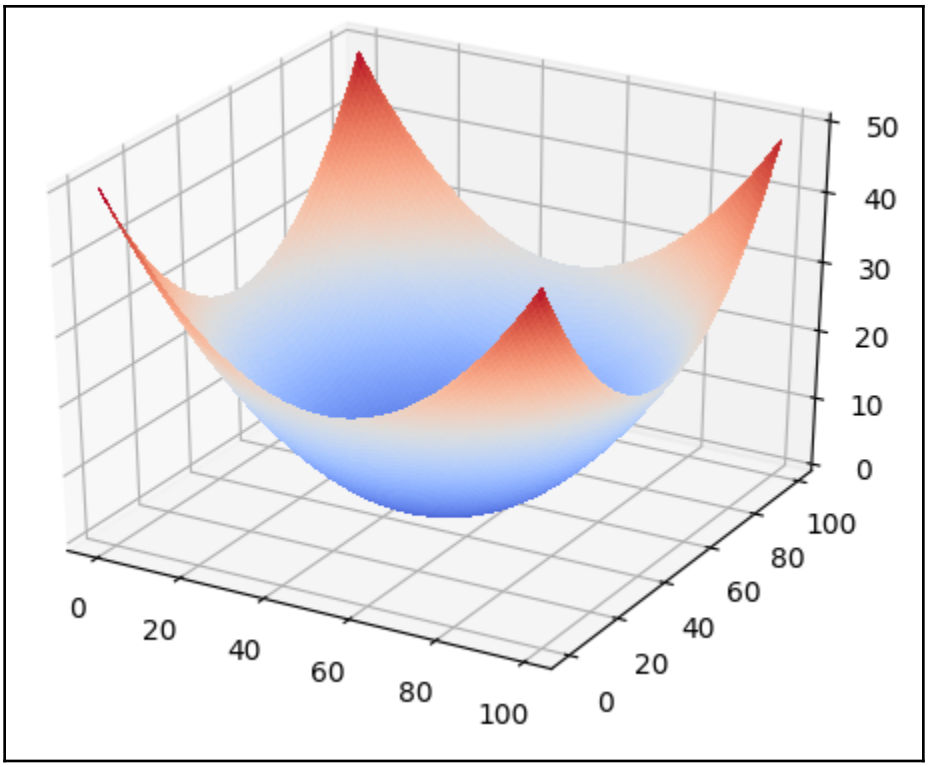

type
0
1
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
x2
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
x1

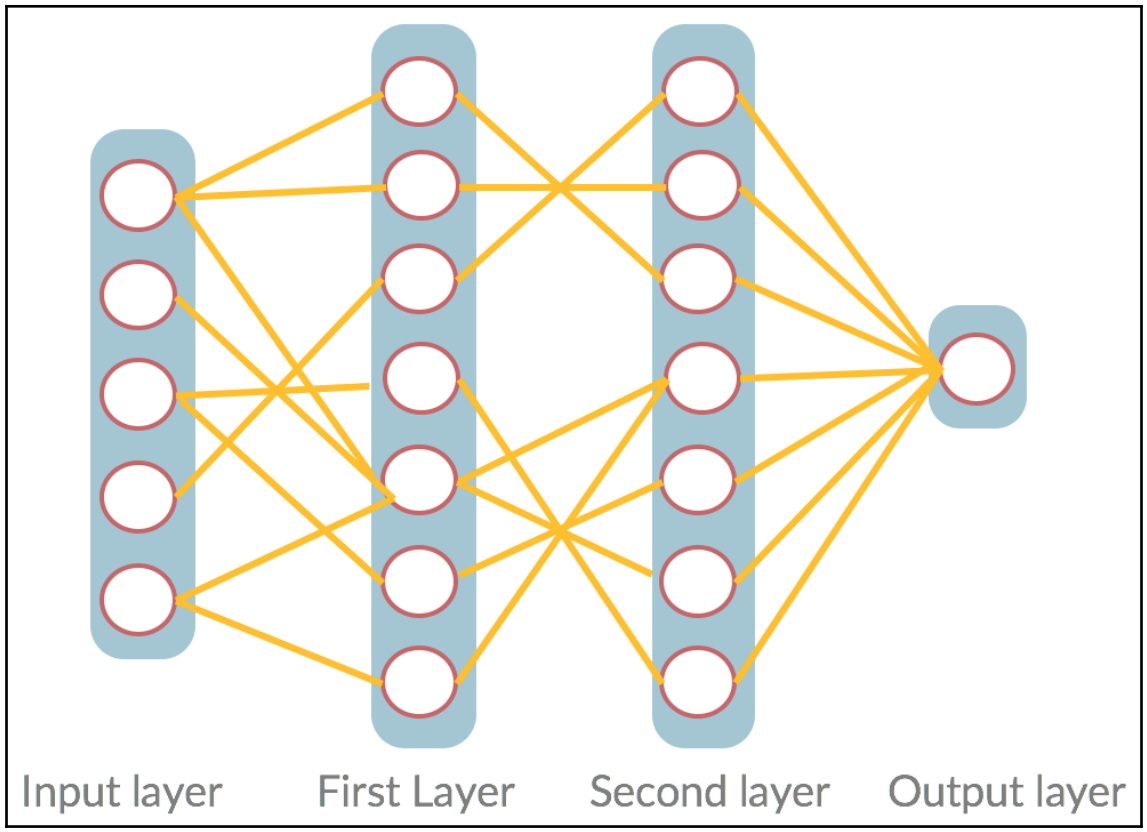

type
0
1
0.6
0.4
0.2
0.0
−0.2
x2
−0.2
0.0
0.2
0.4
0.6
x1

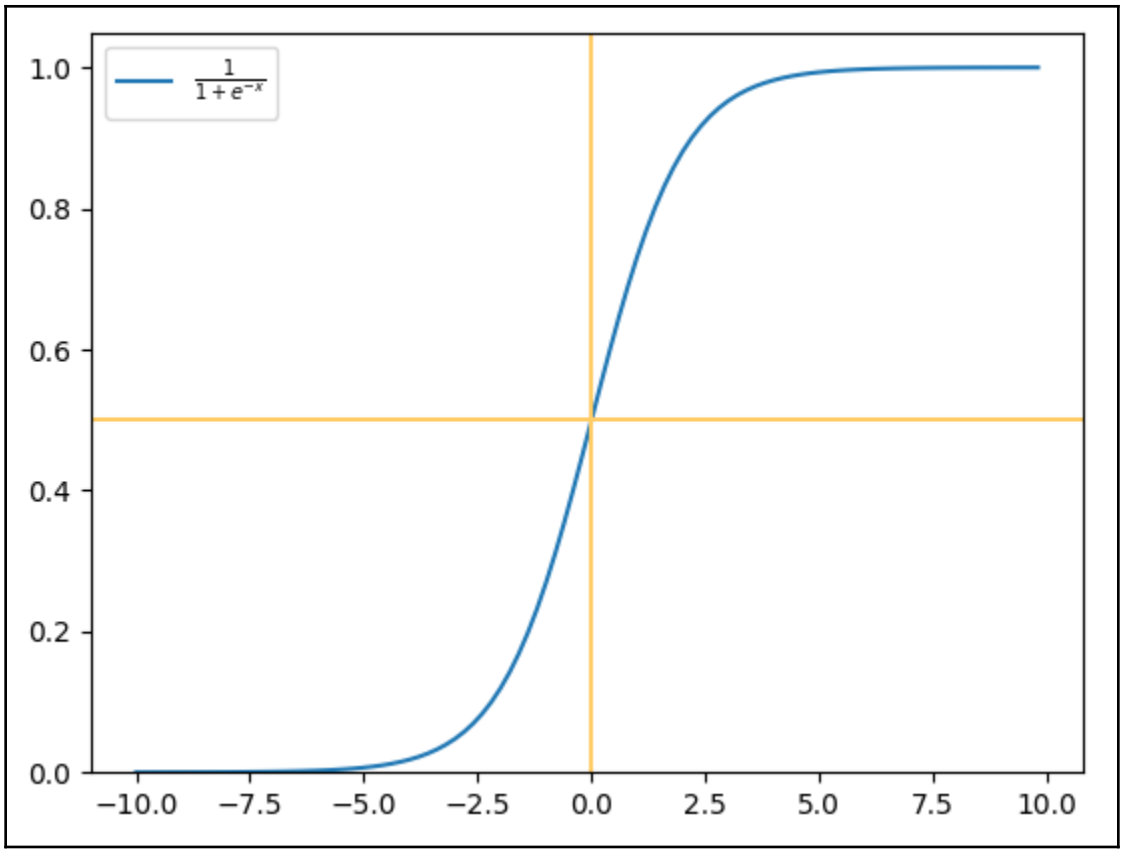

1.0
0.8
0.6
0.4
0.2
0.0
y
0.0
0.2
0.4
0.6
0.8
1.0
x
type
0
1

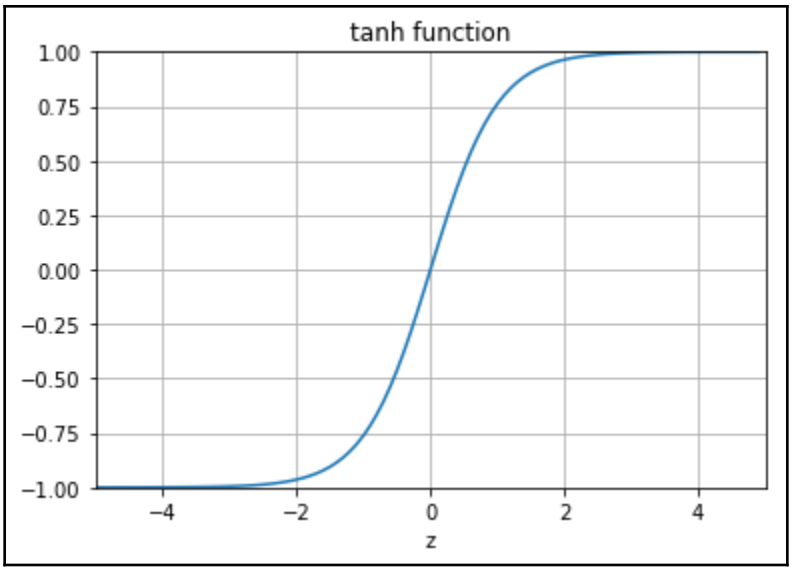

1.0
0.8
0.6
0.4
0.2
0.0
y
0.0
0.2
0.4
0.6
0.8
1.0
x
type
0
1

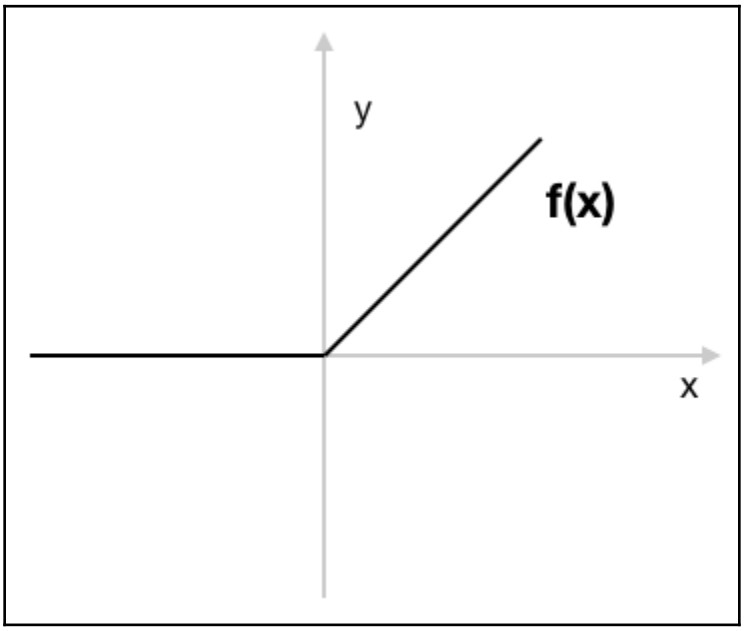

25
20
15
10
5
0
−4
−2
0
2
4

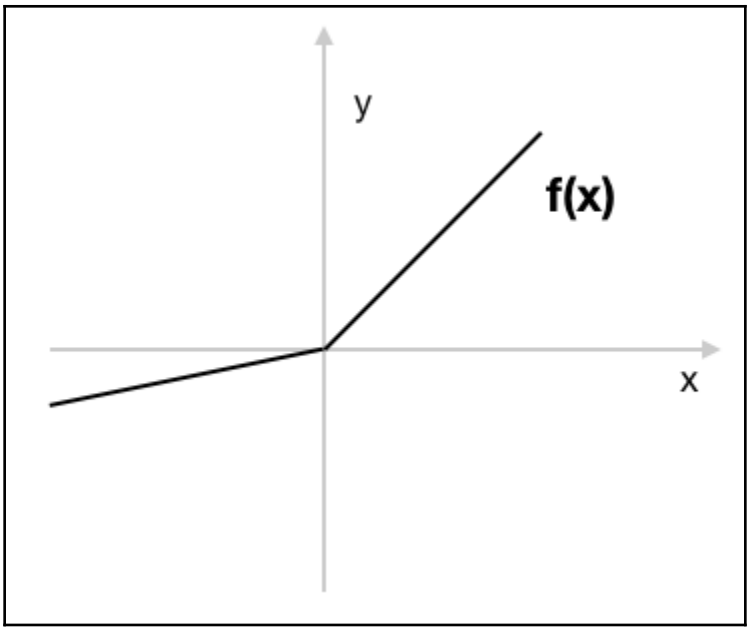

50
40
30
20
10
0
0
20
40
60
80
100
0
20
40
60
80
100

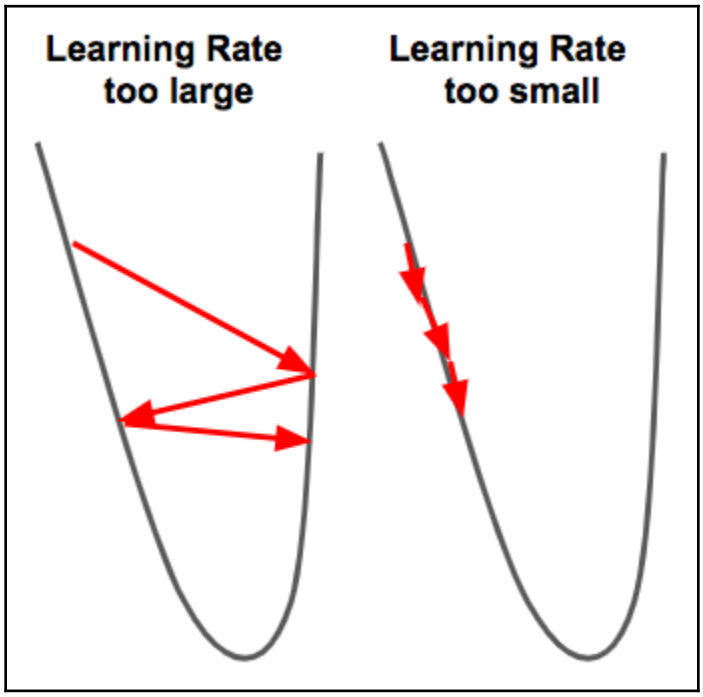

Input layer
First Layer
Second layer
Output layer

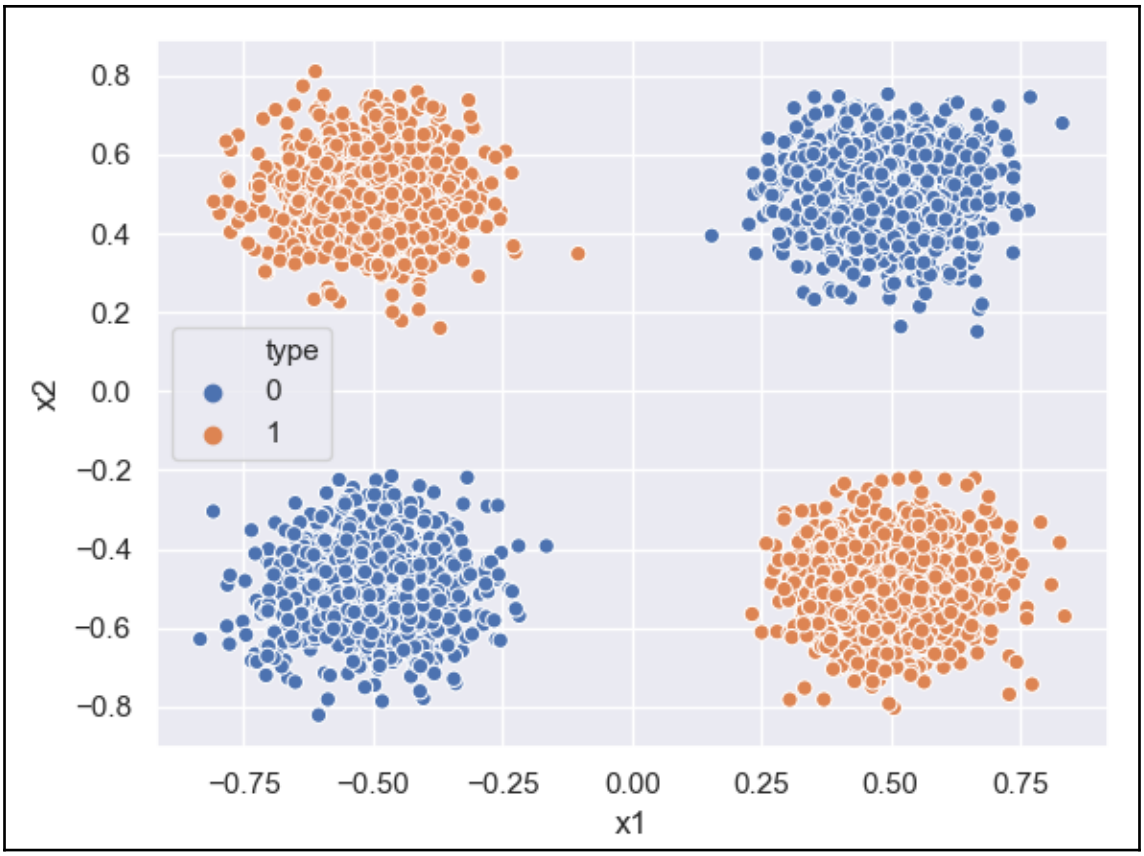

$\frac{1}{1+e^{-x}}$
1.0
0.8
0.6
0.4
0.2
0.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

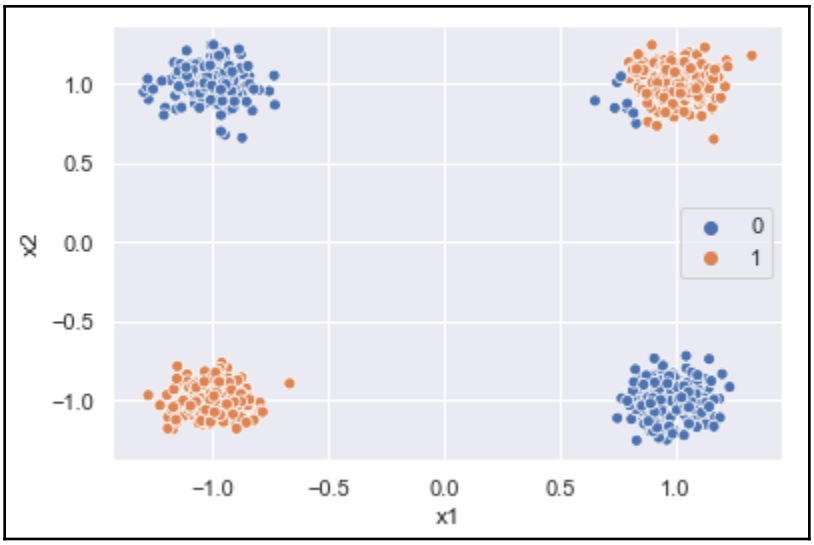

tanh function
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−4
−2
0
2
4
z

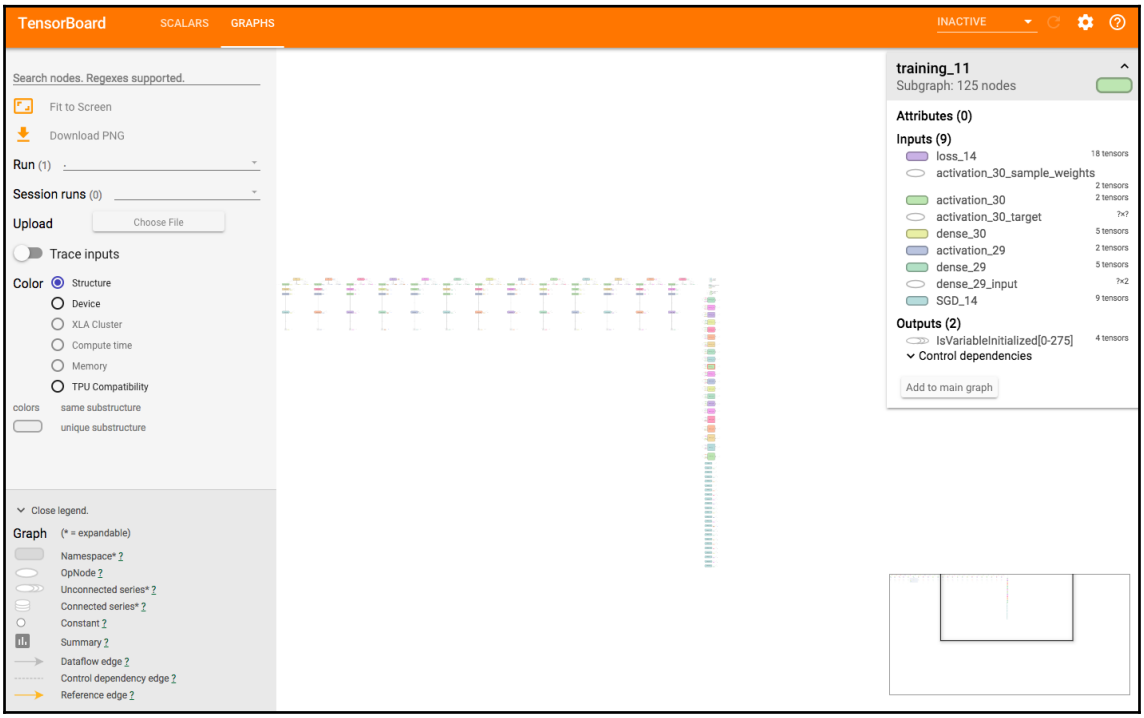

y
f(x)
x

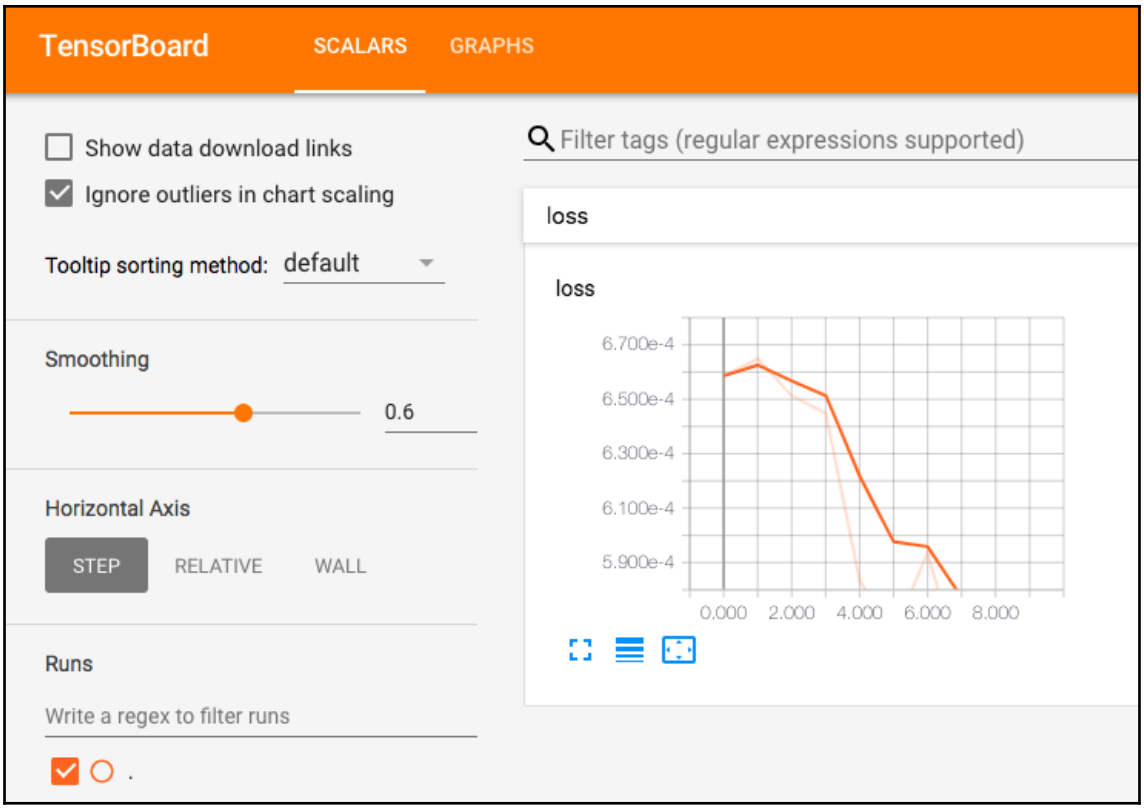

y
f(x)
x


Learning Rate
too large
Learning Rate
too small

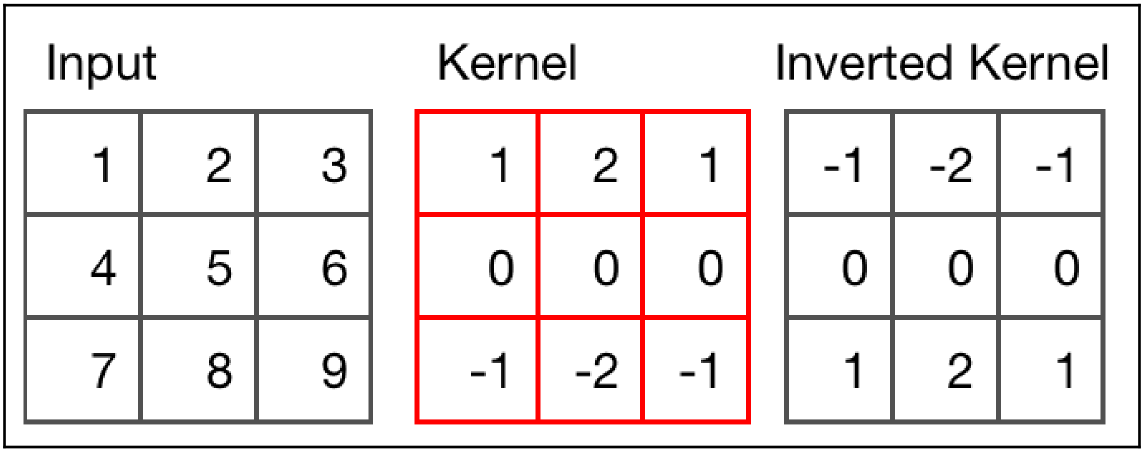

type
0
1
x2
x1
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75


1.0
0.5
0.0
−0.5
−1.0
x2
−1.0
−0.5
0.0
0.5
1.0
x1
0
1

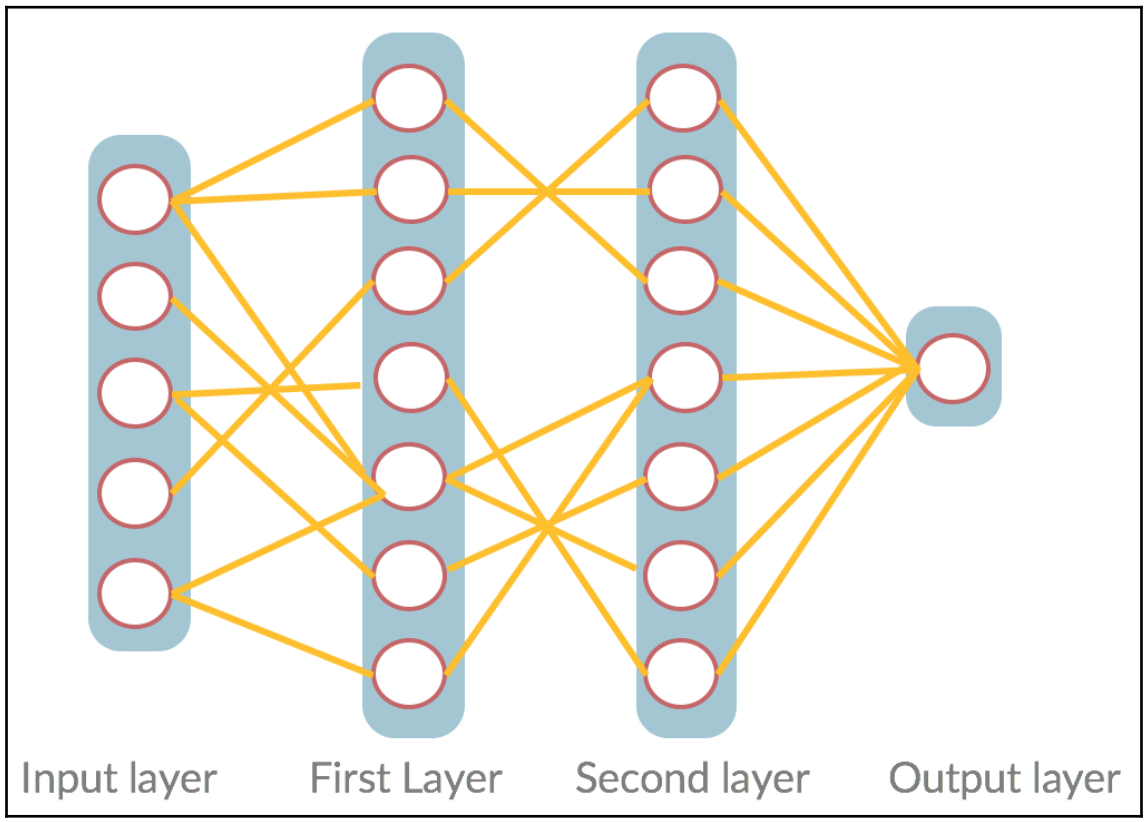

TensorBoard
SCALARS
GRAPHS
INACTIVE
Search nodes. Regexes supported.
Fit to Screen
Download PNG
Run (1)
Session runs (0)
Upload
Choose File
Trace inputs
Color
Structure
Device
XLA Cluster
Compute time
Memory
TPU Compatibility
colors
same substructure
unique substructure
Close legend.
Graph
(* = expandable)
Namespace* ?
OpNode ?
Unconnected series* ?
Connected series* ?
Constant ?
Summary ?
Dataflow edge ?
Control dependency edge ?
Reference edge ?
training_11
Subgraph: 125 nodes
Attributes (0)
Inputs (9)
loss_14
18 tensors
activation_30_sample_weights
2 tensors
activation_30
2 tensors
activation_30_target
?×?
dense_30
5 tensors
activation_29
2 tensors
dense_29
5 tensors
dense_29_input
?×2
SGD_14
9 tensors
Outputs (2)
IsVariableInitialized[0-275]
4 tensors
Control dependencies
Add to main graph

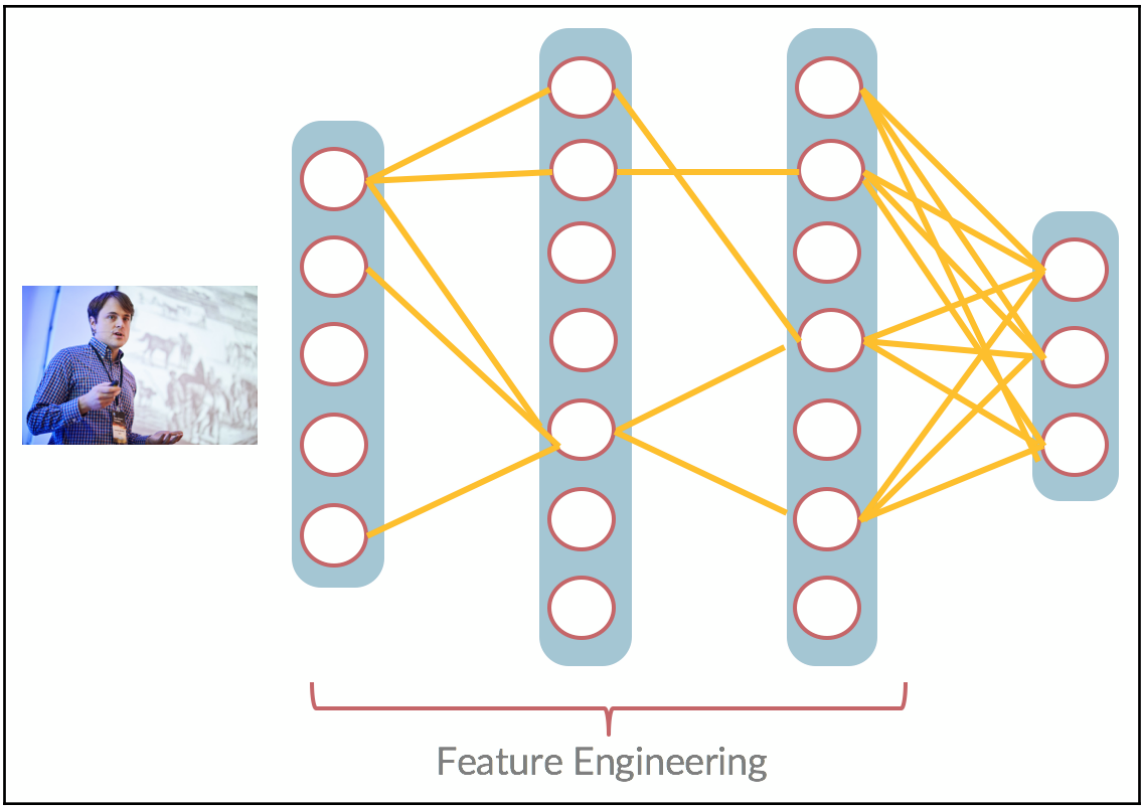

TensorBoard
SCALARS
GRAPHS
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
Filter tags (regular expressions supported)
loss
loss
6.700e-4
6.500e-4
6.300e-4
6.100e-4
5.900e-4
0.000
2.000
4.000
6.000
8.000

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	84	185	159	151	60	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	222	254	254	254	254	241	198	198	198	198	198	198	198	198	170	52	0	0	0	0	0	0
0	0	0	0	0	0	67	114	72	114	163	227	254	225	254	254	254	250	229	254	254	140	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	17	66	14	67	67	67	59	21	236	254	106	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	253	209	18	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	233	255	83	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	129	254	238	44	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59	249	254	62	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	133	254	187	5	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	205	248	58	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	126	254	182	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	251	240	57	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	19	221	254	166	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	3	203	254	219	35	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	38	254	254	77	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	31	224	254	115	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	133	254	254	52	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	61	242	254	254	52	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	121	254	254	219	40	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	121	254	207	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



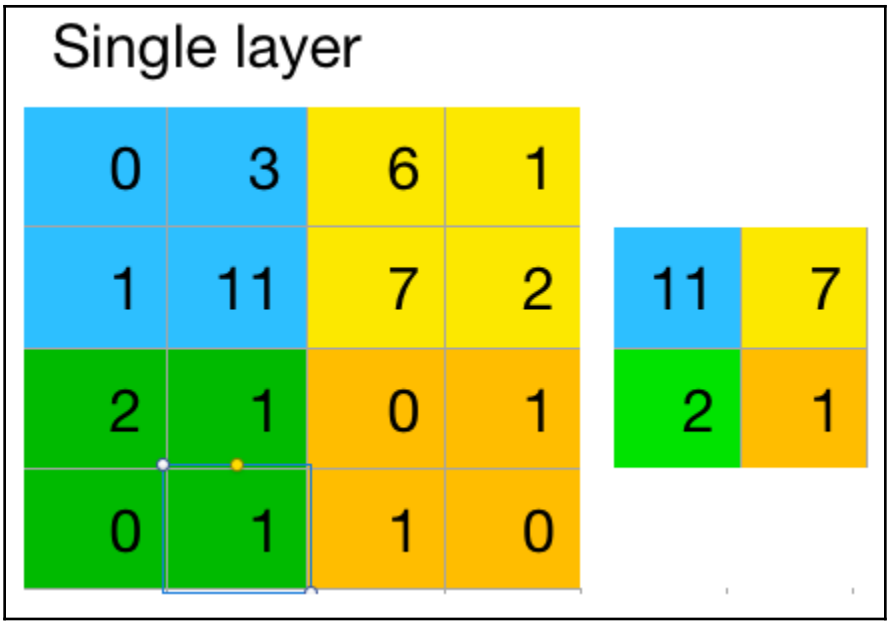
Input			Kernell	
a	b	c	w11	w12
d	e	f	w21	w22
g	h	i		
aw11+bw12+dw21+ew22	bw11+cw12+ew21+fw22			

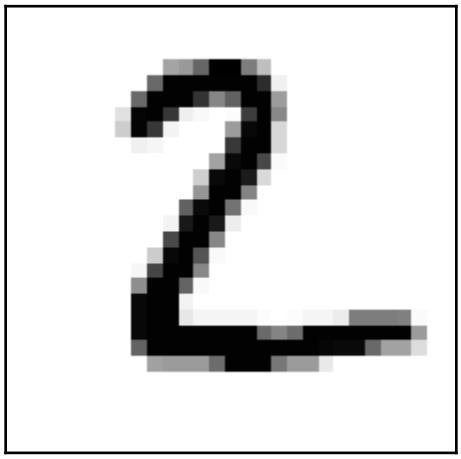


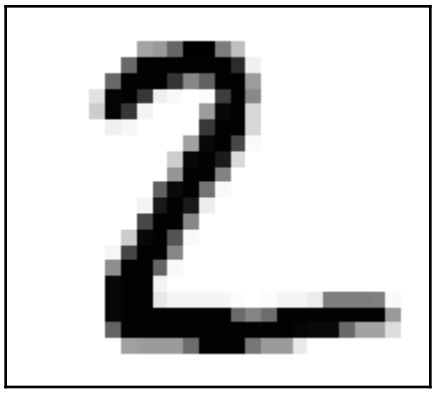

Feature Engineering

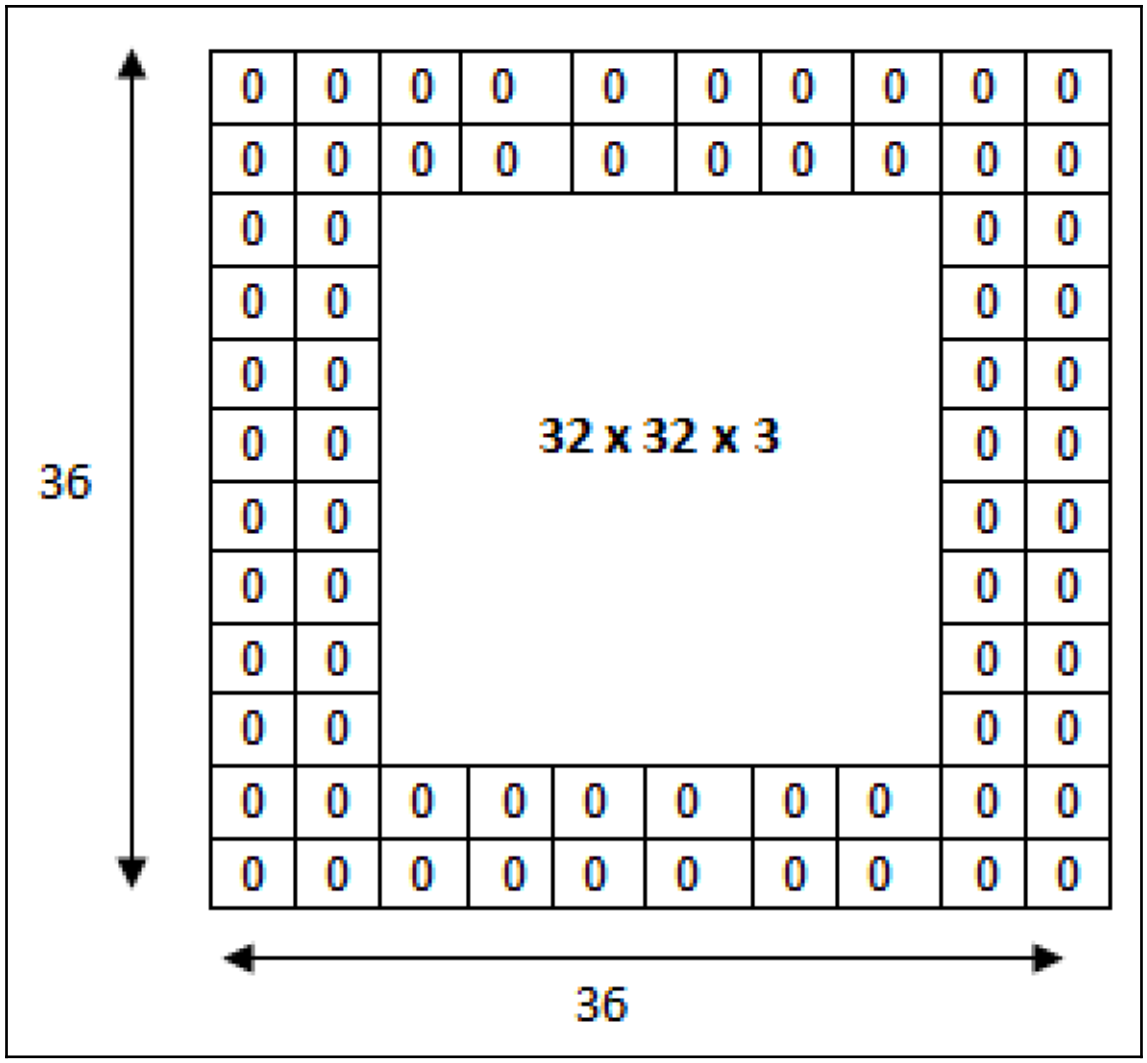
0	0	0	0	0	40	40
0	0	0	0	80	40	40
0	0	0	90	80	0	0
0	0	90	80	0	0	0
0	0	90	90	0	0	0
0	0	90	80	0	0	0
0	0	90	80	0	0	0

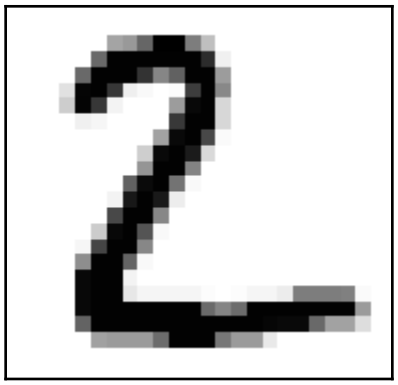
0	0	0	0	0	10	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0


Single layer
0 3 6 1
1 11 7 2
2 1 0 1
0 1 1 0
11 7
2 1

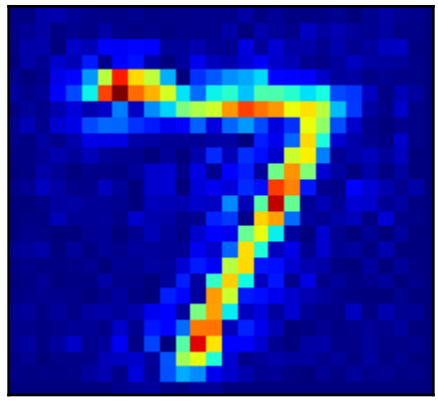


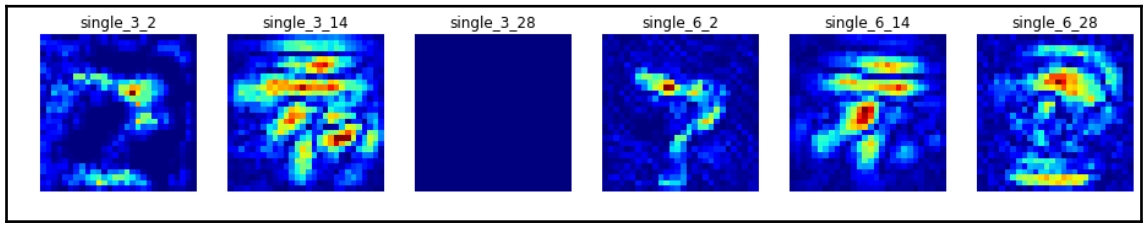


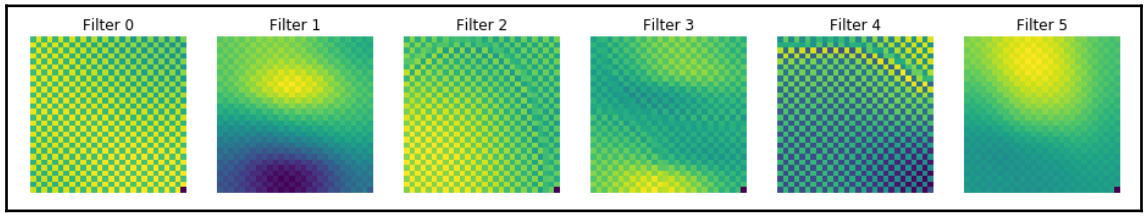

0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
32 x 32 x 3
36
36

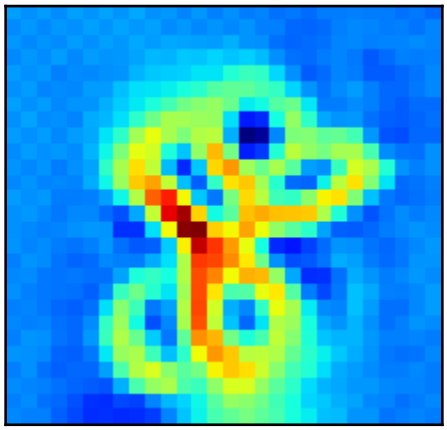



raw
fer2013.csv
PrivateTest
00_Anger
01_Disgust
02_Fear
03_Happy
04_Neutral
05_Sad
06_Surprise
PublicTest
00_Anger
01_Disgust
02_Fear
03_Happy
04_Neutral
05_Sad
06_Surprise
Training
00_Anger
01_Disgust
02_Fear
03_Happy
04_Neutral
05_Sad
06_Surprise

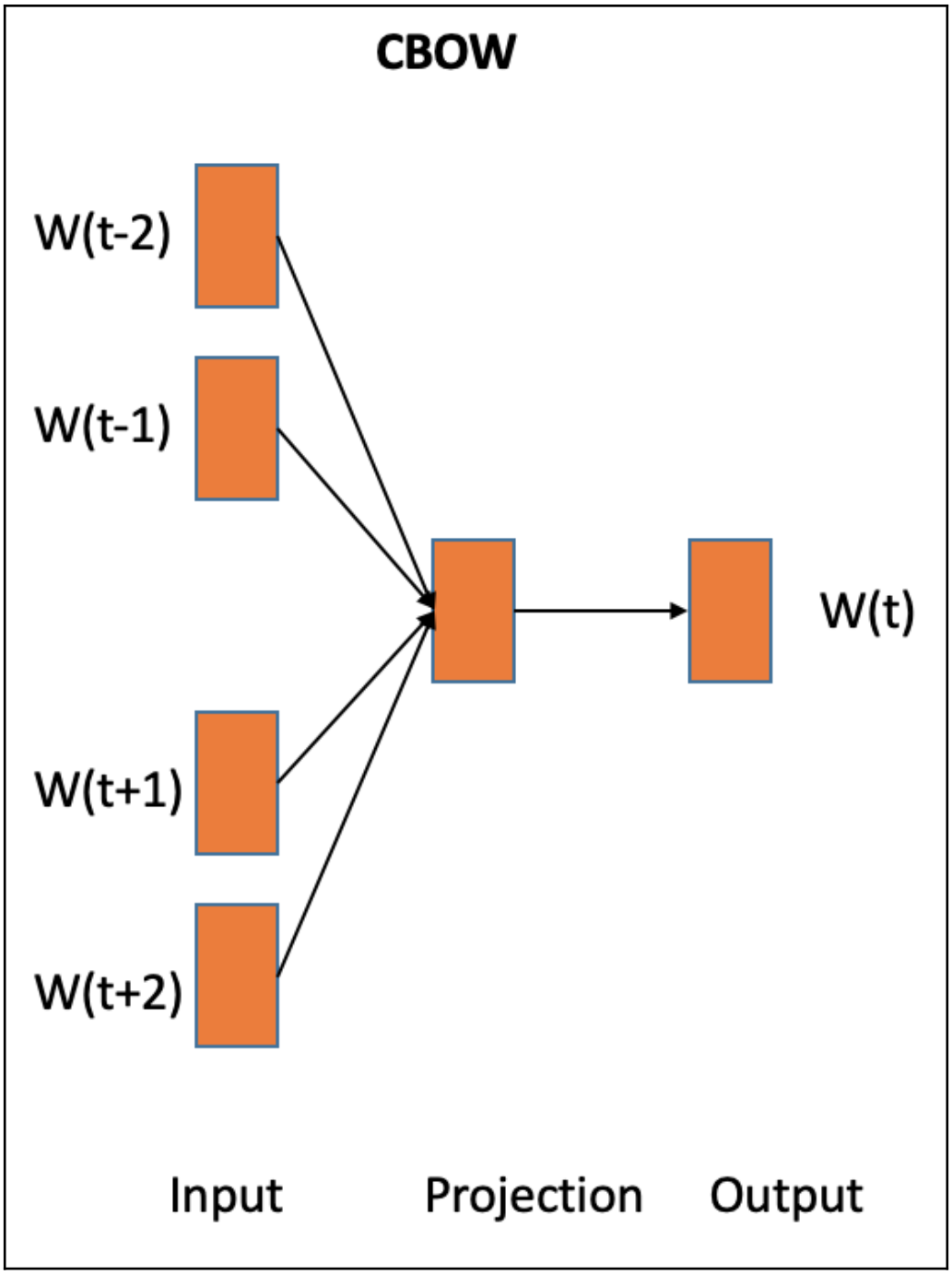


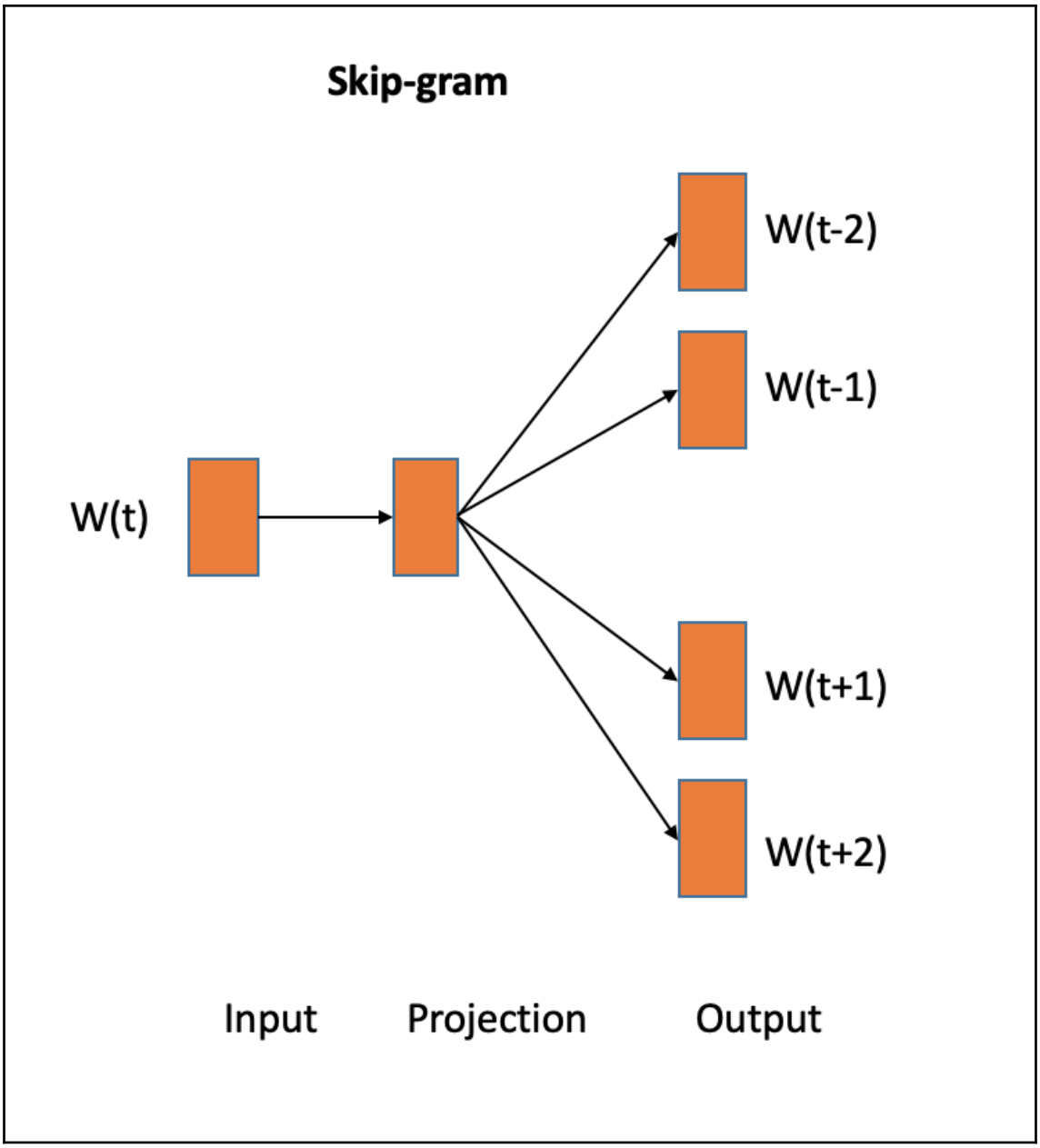

single_3_2
single_3_14
single_3_28
single_6_2
single_6_14
single_6_28


Filter 0
Filter 1
Filter 2
Filter 3
Filter 4
Filter 5



Chapter 4: Exploiting Text Embedding


CBOW
W(t-2)
W(t-1)
W(t+1)
W(t+2)
W(t)
Input
Projection
Output

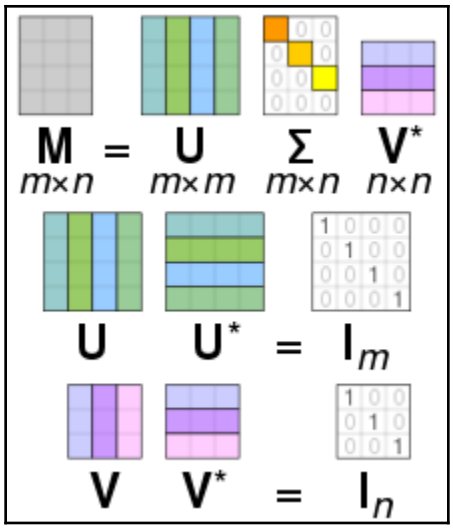

Skip-gram
W(t-2)
W(t-1)
W(t)
W(t+1)
W(t+2)
Input
Projection
Output

	text	sentiment
0	This is good pizza	1
1	I love Italian pizza	1
2	The best pizza	1
3	nice pizza	1
4	Excellent pizza	1
5	I love pizza	1
6	The pizza was alright	0
7	disgusting pineapple pizza	0
8	not good pizza	0
9	bad pizza	0
10	very bad pizza	0
11	I had better pizza	0

```
[[15, 18, 19, 2],
 [15, 12, 7, 2],
 [11, 15, 2],
 [4, 2],
 [15, 2],
 [15, 12, 2],
 [11, 2, 2, 11],
 [4, 7, 2],
 [9, 19, 2],
 [19, 2],
 [6, 19, 2],
 [15, 1, 14, 2]]
```

```
[[15 18 19  2  0]
 [15 12  7  2  0]
 [11 15  2  0  0]
 [ 4  2  0  0  0]
 [15  2  0  0  0]
 [15 12  2  0  0]
 [11  2  2 11  0]
 [ 4  7  2  0  0]
 [ 9 19  2  0  0]
 [19  2  0  0  0]
 [ 6 19  2  0  0]
 [15  1 14  2  0]]
```

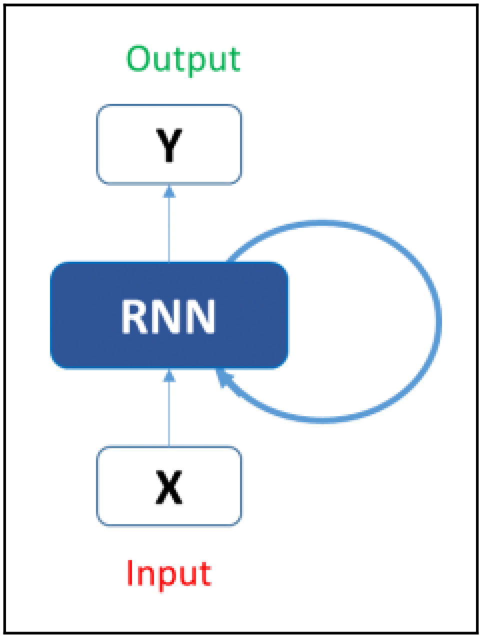
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_4 (Embedding)      (None, 5, 8)              160
_________________________________________________________________
flatten_4 (Flatten)          (None, 40)                0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 41
=================================================================
Total params: 201
Trainable params: 201
Non-trainable params: 0
_________________________________________________________________
None
Accuracy: 83.333331
```

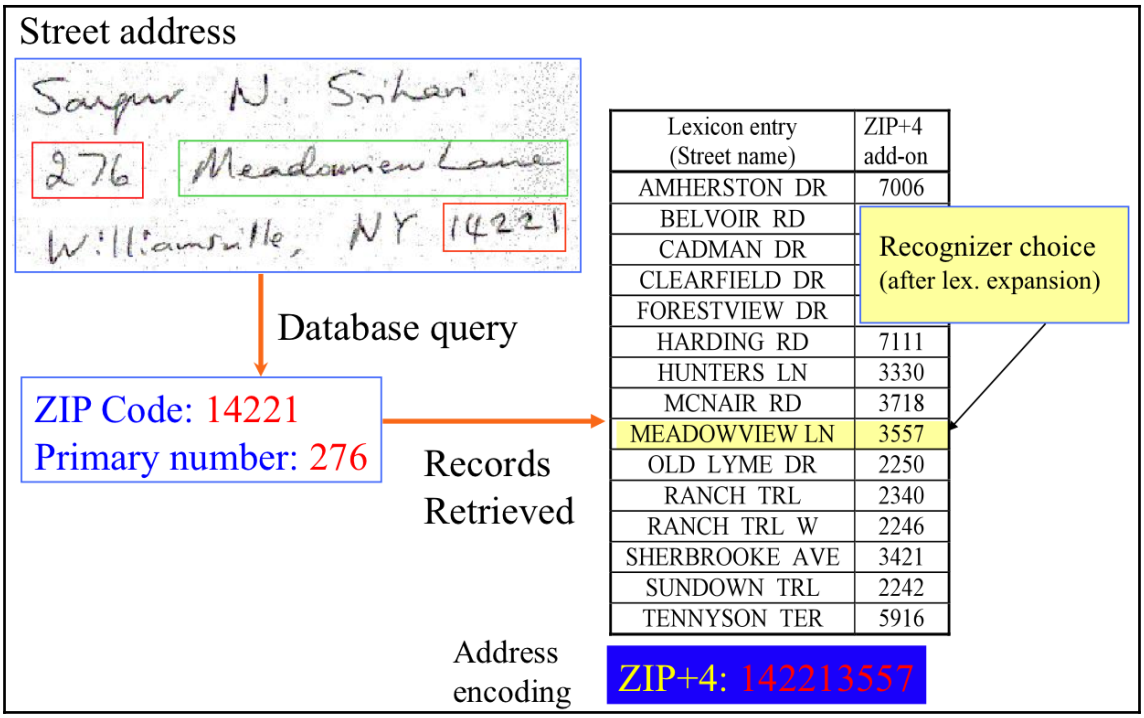


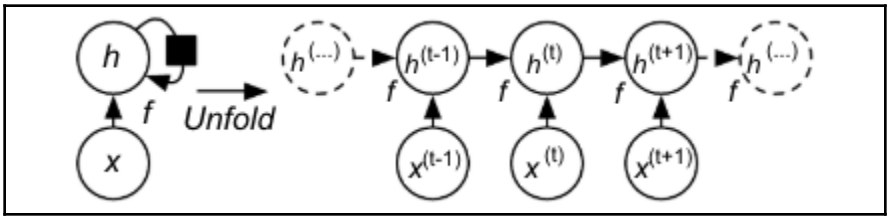
Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

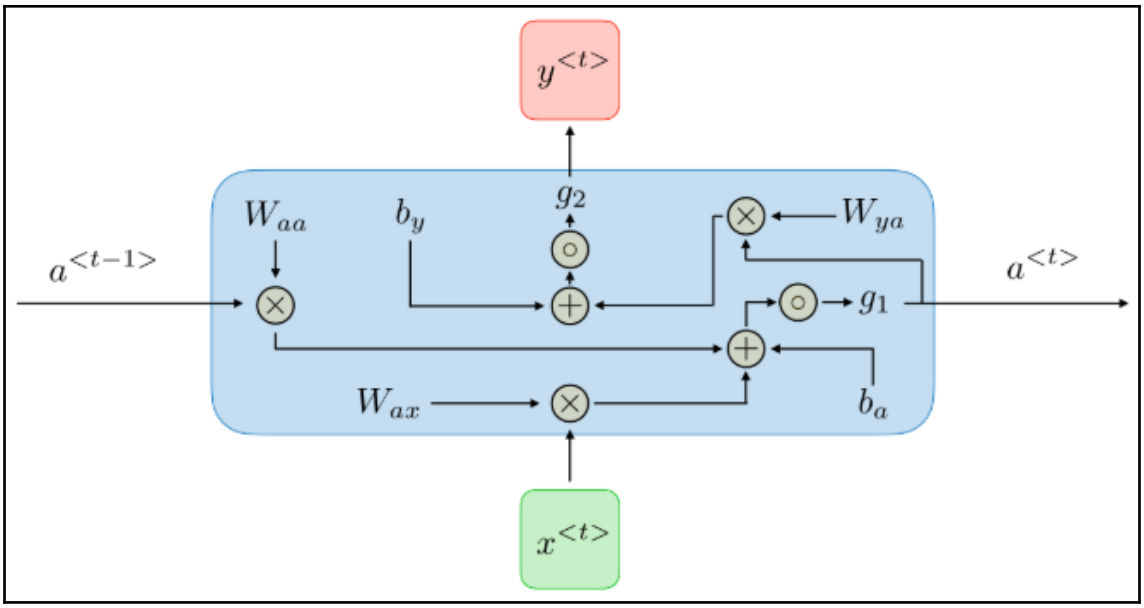
Chapter 5: Working with RNNs

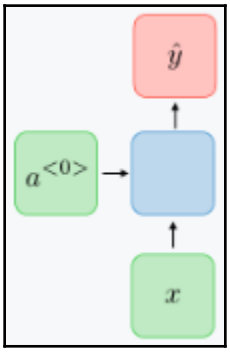
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	84	185	159	151	60	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	222	254	254	254	254	241	198	198	198	198	198	198	198	198	170	52	0	0	0	0	0	0
0	0	0	0	0	0	67	114	72	114	163	227	254	225	254	254	254	250	229	254	254	140	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	17	66	14	67	67	67	59	21	236	254	106	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	253	209	18	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	233	255	83	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	129	254	238	44	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59	249	254	62	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	133	254	187	5	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	205	248	58	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	126	254	182	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	251	240	57	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	19	221	254	166	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	3	203	254	219	35	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	38	254	254	77	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	31	224	254	115	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	133	254	254	52	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	61	242	254	254	52	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	121	254	254	219	40	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	121	254	207	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

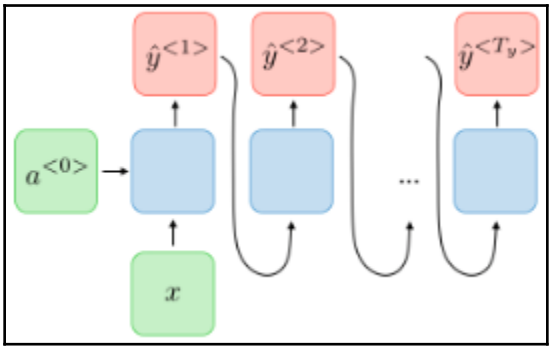


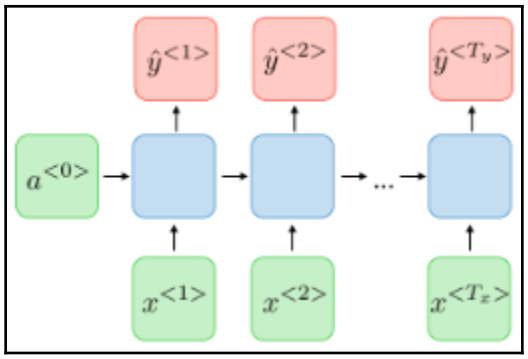

Street address
Sapur N. Srihari
276
Meadowview Lane
Williamsville, NY 14221
Database query
ZIP Code: 14221
Primary number: 276
Records Retrieved
Address encoding
ZIP+4: 142213557
Recognizer choice (after lex. expansion)
Lexicon entry (Street name)
ZIP+4 add-on
AMHERSTON DR 7006
BELVOIR RD
CADMAN DR
CLEARFIELD DR
FORESTVIEW DR
HARDING RD 7111
HUNTERS LN 3330
MCNAIR RD 3718
MEADOWVIEW LN 3557
OLD LYME DR 2250
RANCH TRL 2340
RANCH TRL W 2246
SHERBROOKE AVE 3421
SUNDOWN TRL 2242
TENNYSON TER 5916

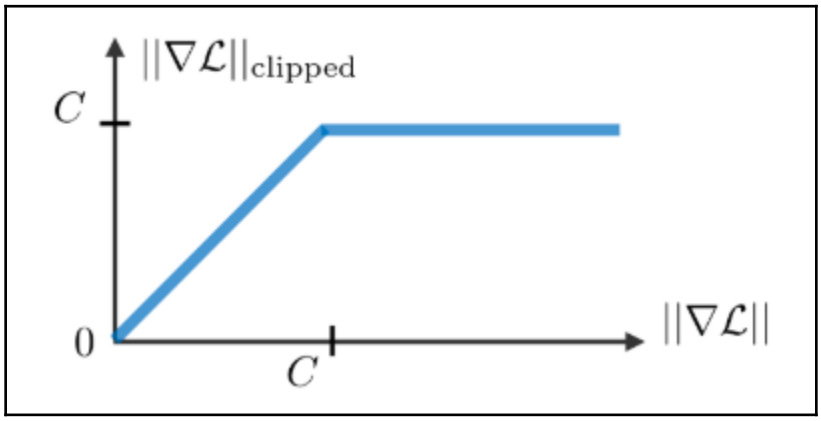

h
f
x
Unfold
h(...)
h(t-1)
h(t)
h(t+1)
x(t-1)
x(t)
x(t+1)

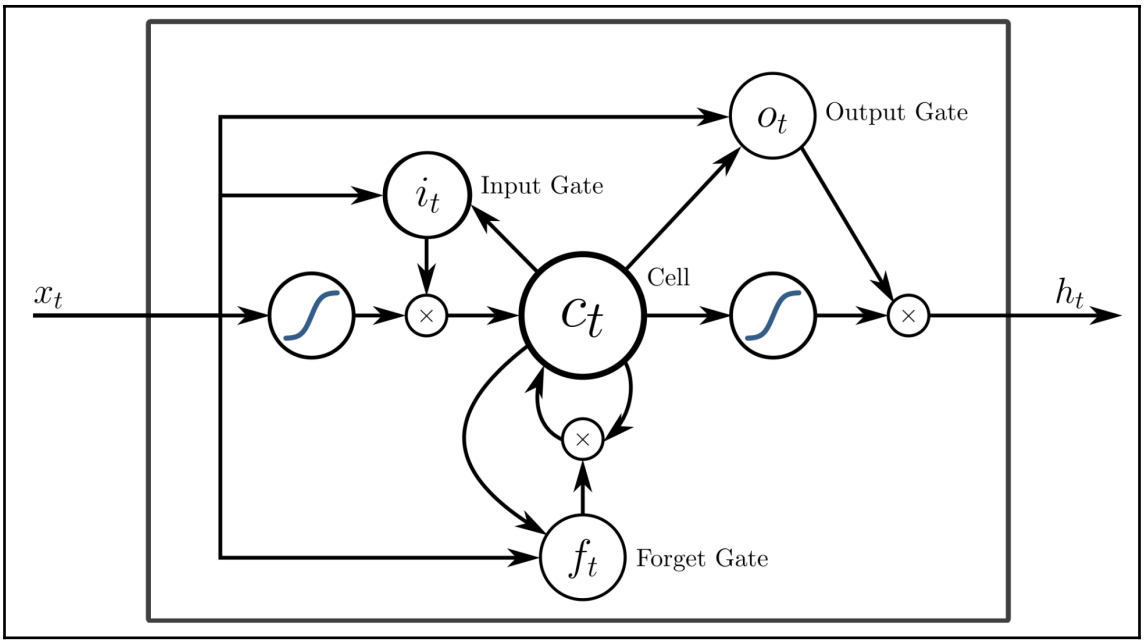

$y^{<t>}$
W_{aa}
b_y
g_2
W_{ya}
$a^{<t-1>}$
$a^{<t>}$
g_1
W_{ax}
b_a
$x^{<t>}$


$\hat{y}$
$a^{<0>}$
x

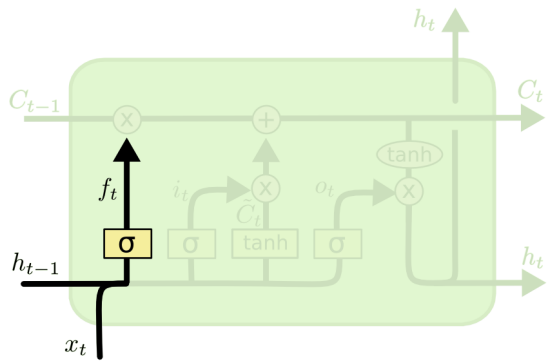

$\hat{y}^{<1>}$
$\hat{y}^{<2>}$
$\hat{y}^{<T_y>}$
$a^{<0>}$
...
x


$\hat{y}^{<1>}$
$\hat{y}^{<2>}$
$\hat{y}^{<T_y>}$
$a^{<0>}$
...
$x^{<1>}$
$x^{<2>}$
$x^{<T_x>}$

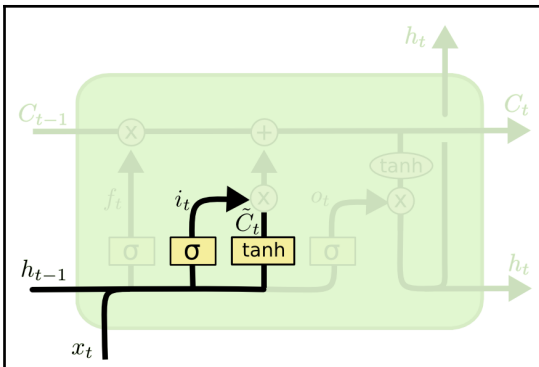

$||\nabla \mathcal{L}||_{\text{clipped}}$
C
0
C
$||\nabla \mathcal{L}||$



$x_t \in \mathbb{R}^a$: input vector to the LSTM unit
$f_t \in \mathbb{R}^h$: forget gate's activation vector
$i_t \in \mathbb{R}^h$: input gate's activation vector
$o_t \in \mathbb{R}^h$: output gate's activation vector
$h_t \in \mathbb{R}^h$: hidden state vector also known as output vector of the LSTM unit
$c_t \in \mathbb{R}^h$: cell state vector
$W \in \mathbb{R}^{h \times d}$, $U \in \mathbb{R}^{h \times h}$ and $b \in \mathbb{R}^h$: weight matrices and bias vector parameters which need to be learned during training

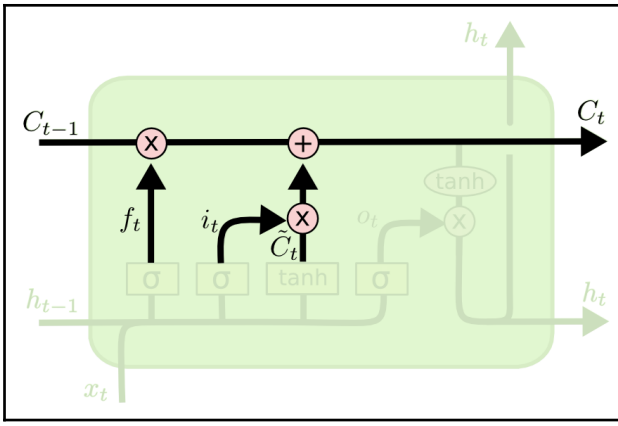


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

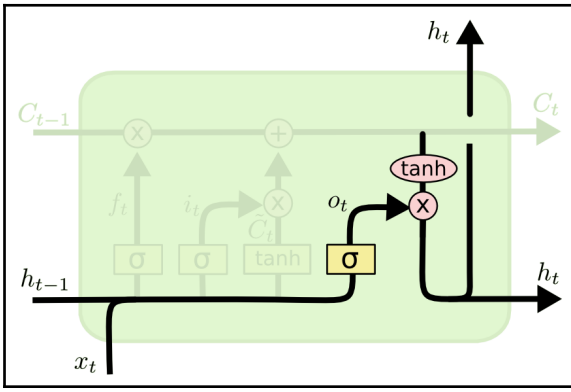


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

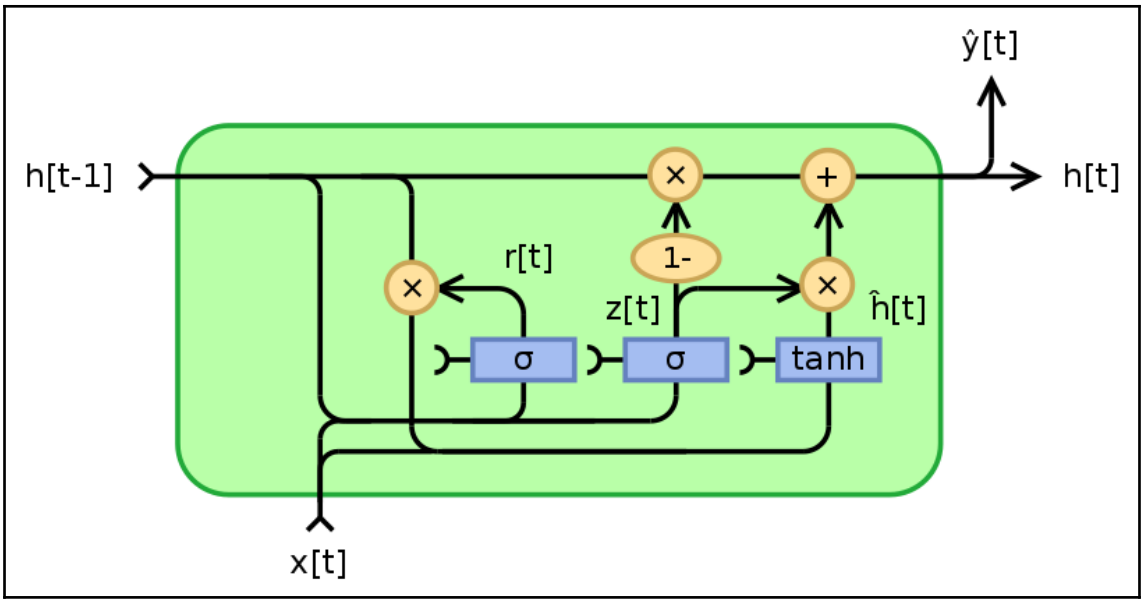


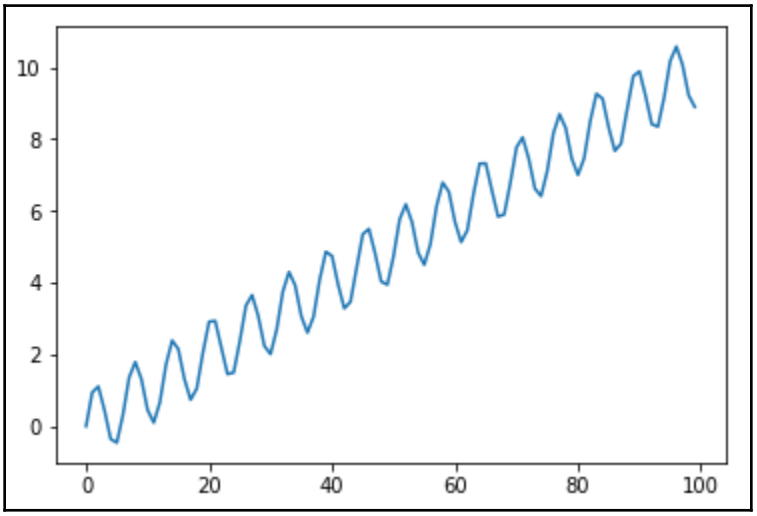
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

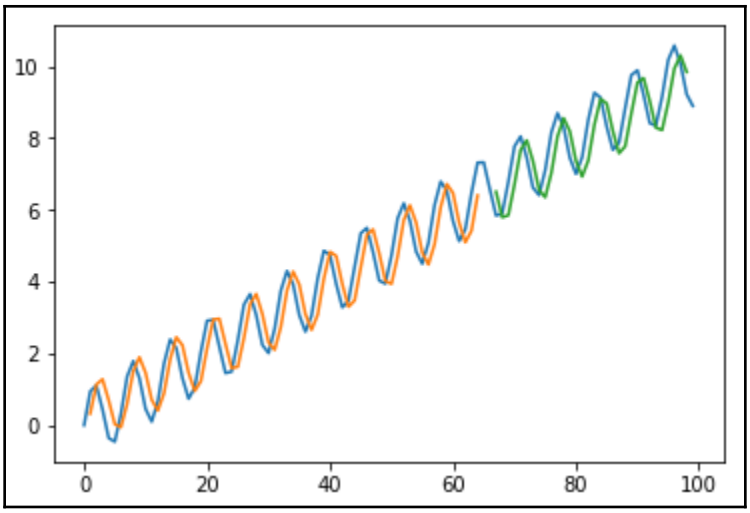


$$o_t = \sigma\left(W_o \; [\, h_{t-1}, x_t] \; + \; b_o\right)$$

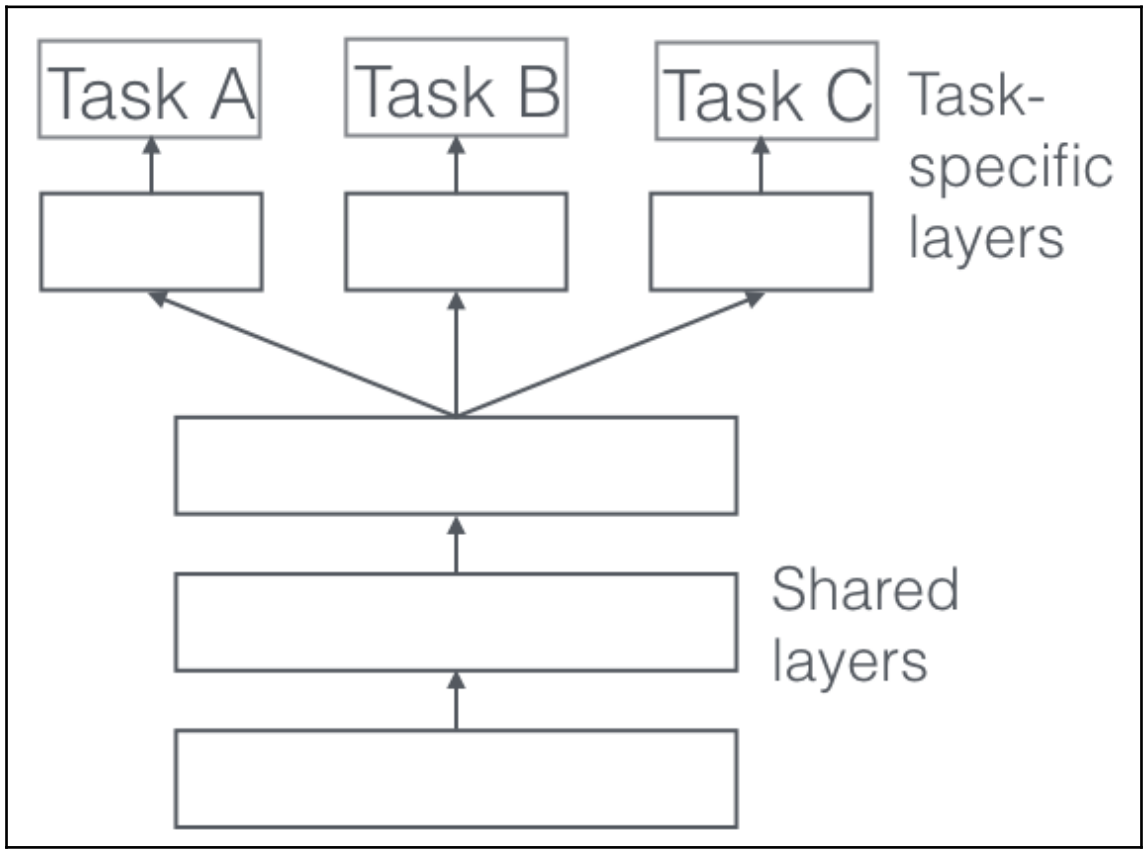
$$h_t = o_t * \tanh\left(C_t\right)$$

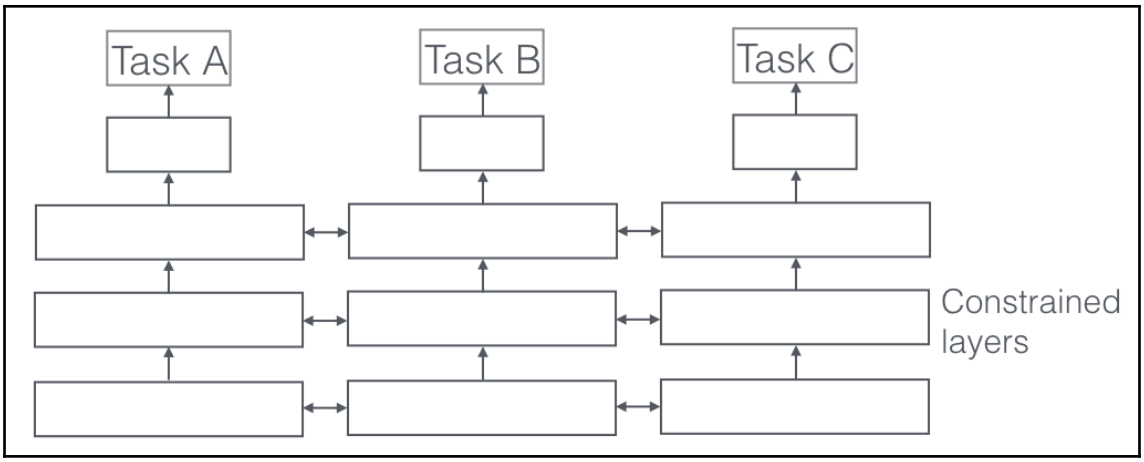

ŷ[t]
h[t-1]
h[t]
×
+
1-
r[t]
z[t]
ĥ[t]
σ
σ
tanh
x[t]

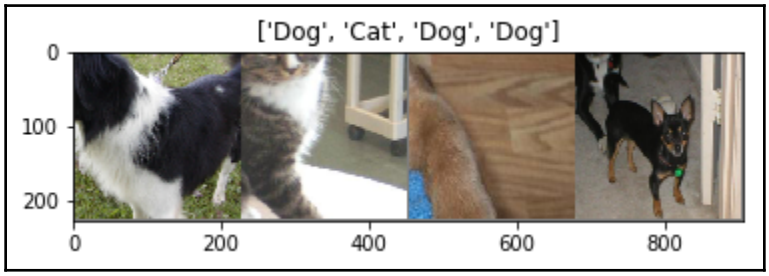

10
8
6
4
2
0
0
20
40
60
80
100


10
8
6
4
2
0
0
20
40
60
80
100

Chapter 6: Reusing Neural Networks with Transfer Learning




Task A
Task B
Task C
Constrained layers


['Dog', 'Cat', 'Dog', 'Dog']
0
100
200
0
200
400
600
800

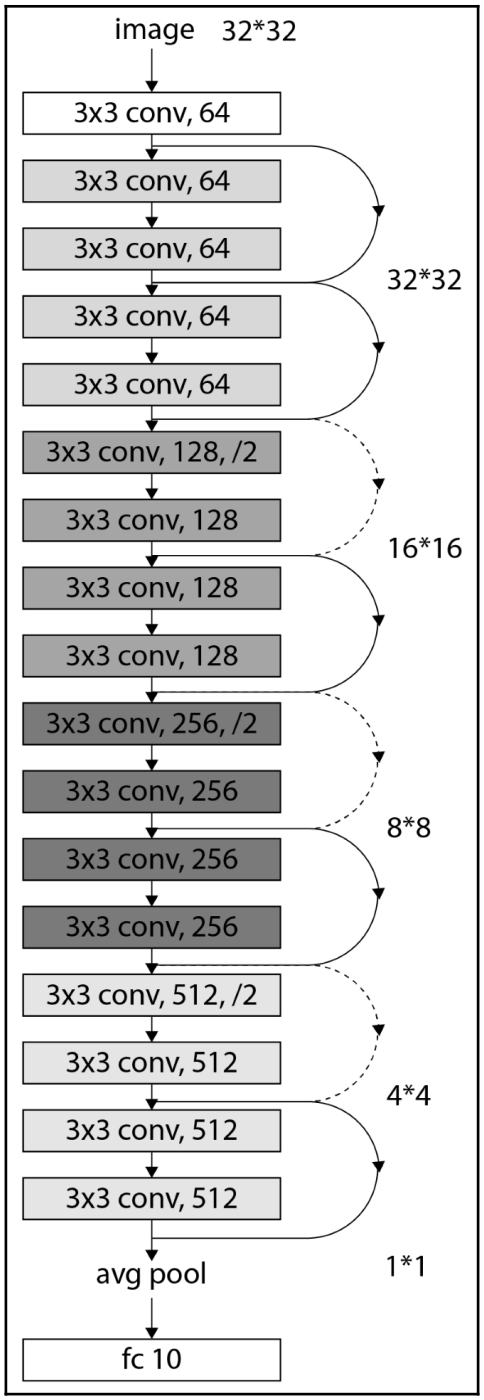
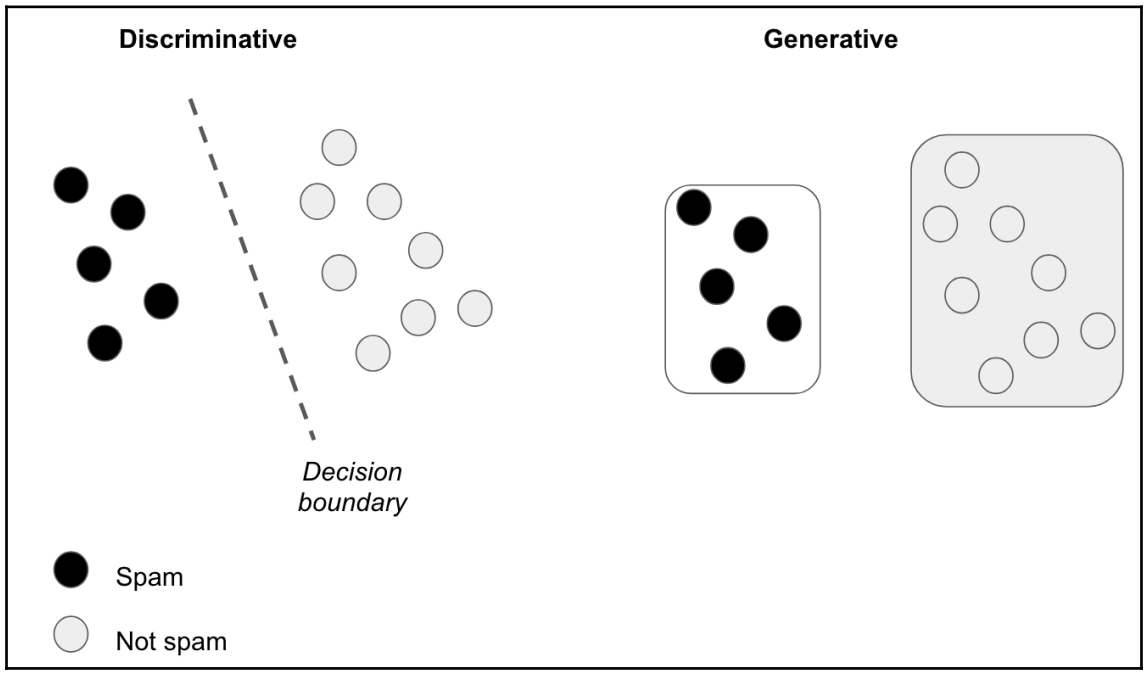
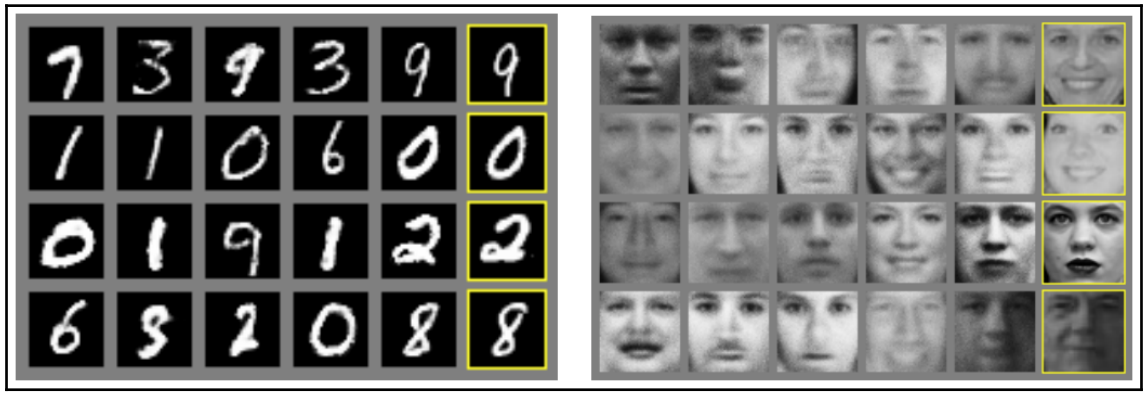
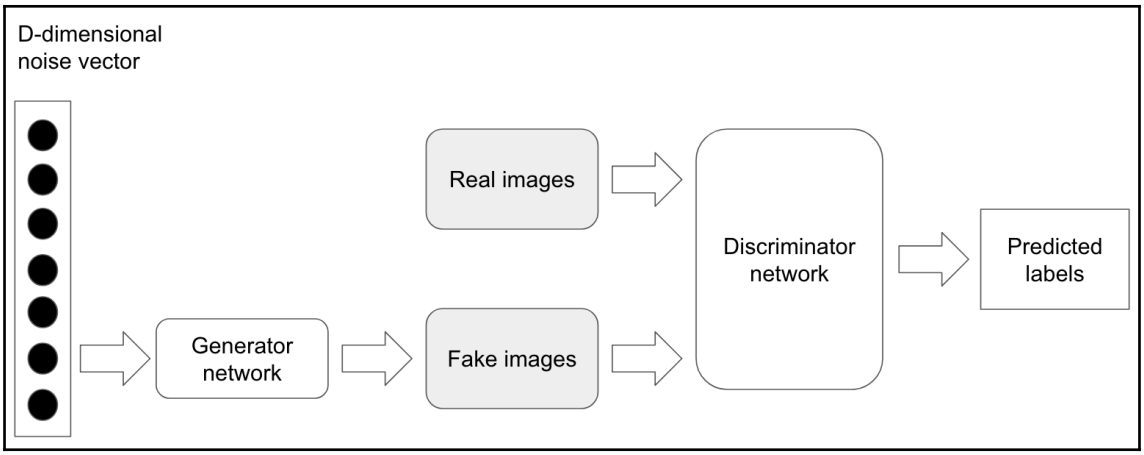


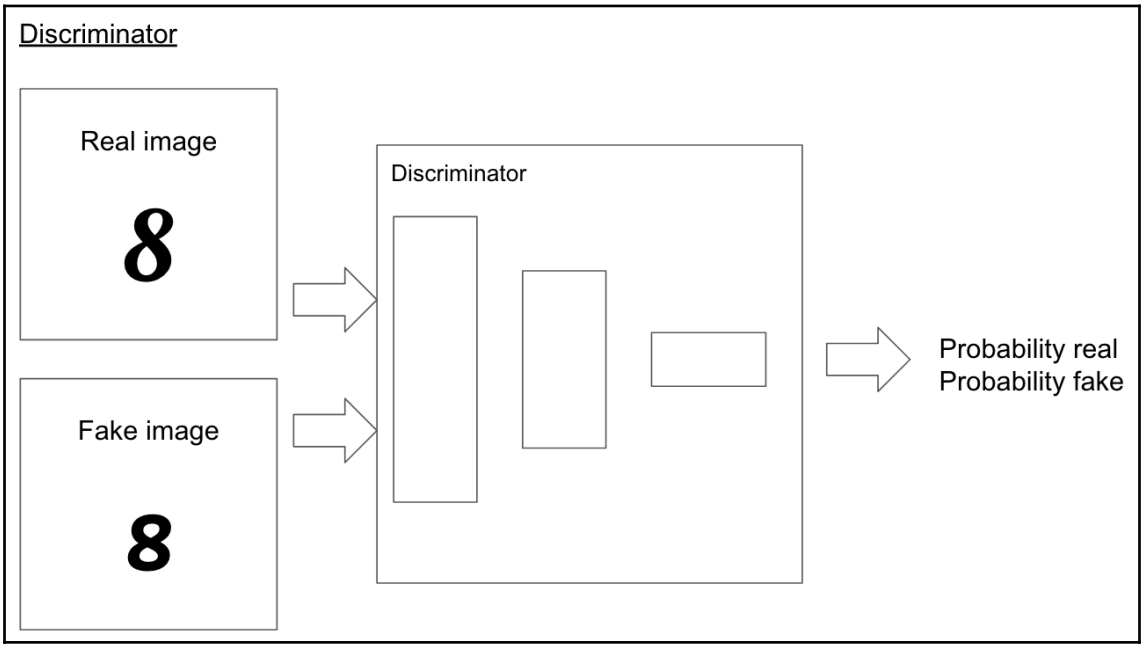
image 32*32
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
avg pool
fc 10
32*32
16*16
8*8
4*4
1*1

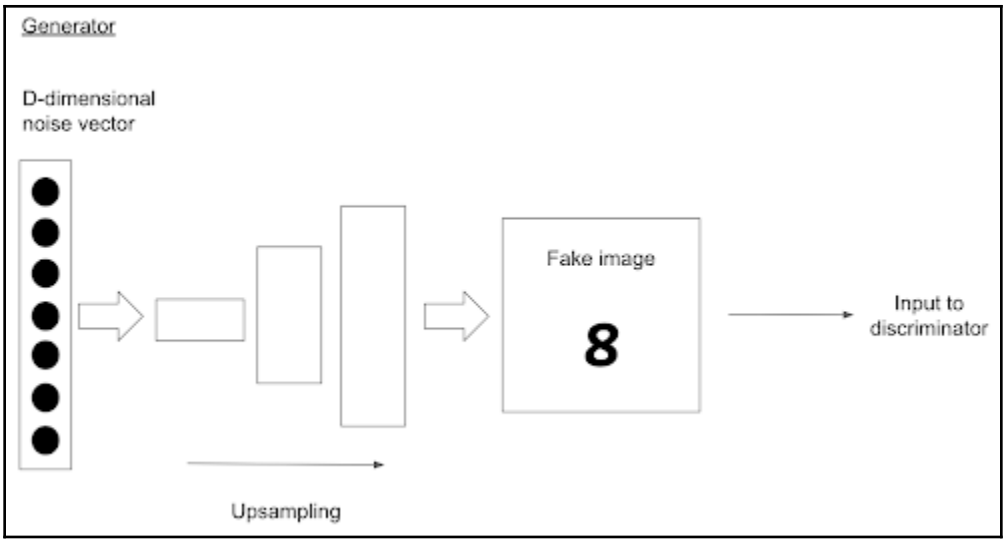
Chapter 7: Working with Generative Algorithms

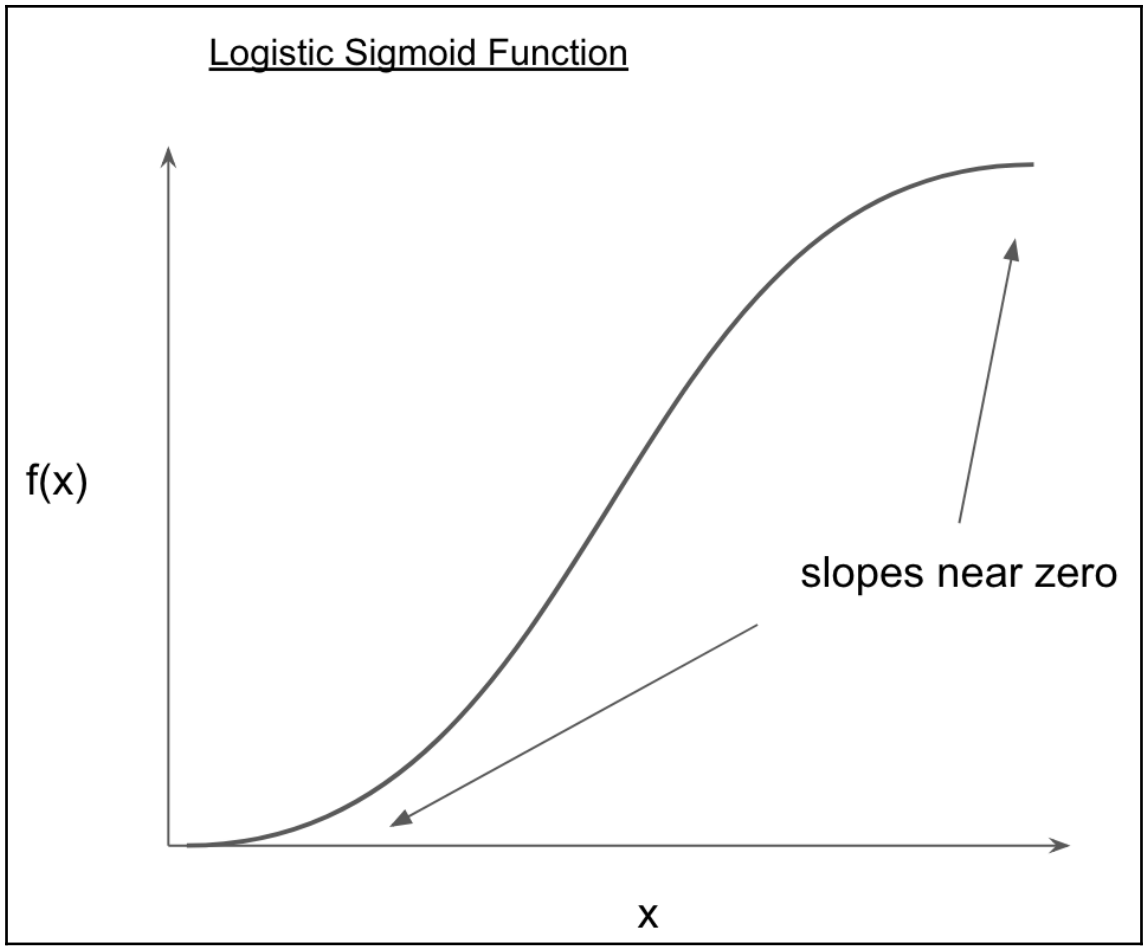


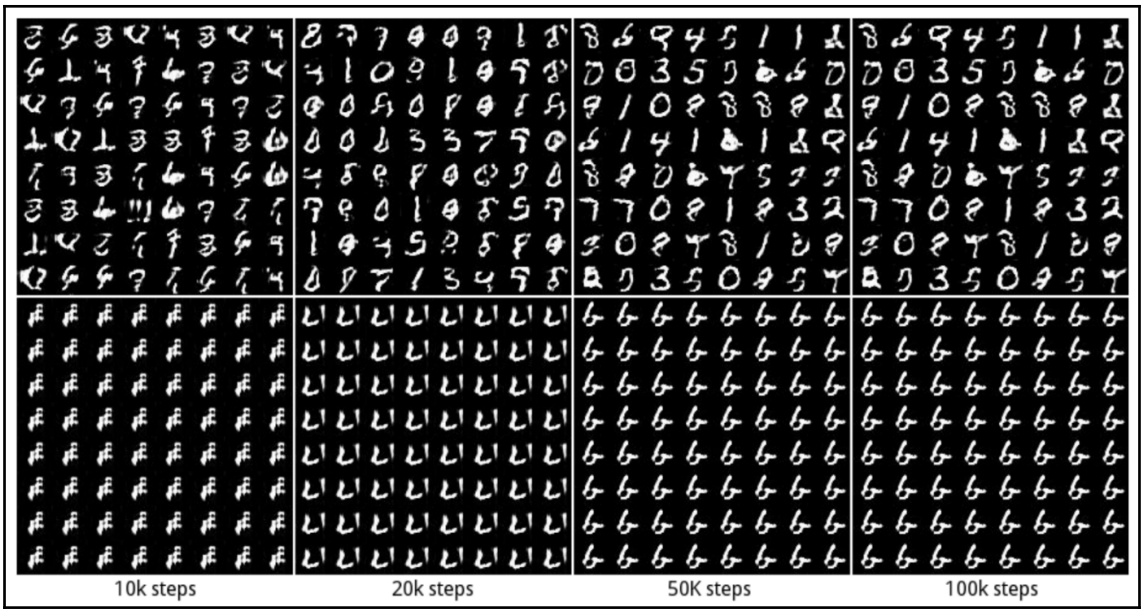


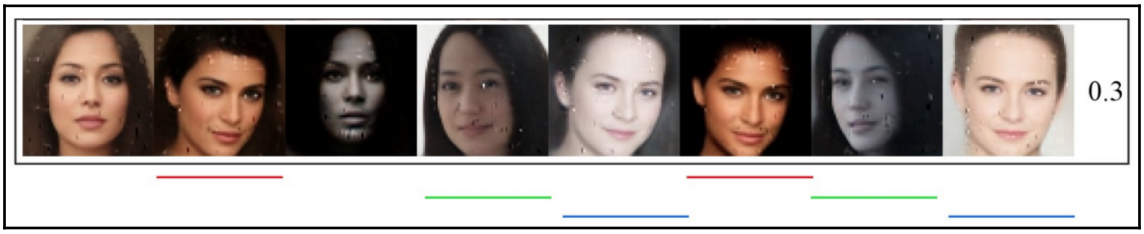

D-dimensional
noise vector
Generator
network
Real images
Fake images
Discriminator
network
Predicted
labels

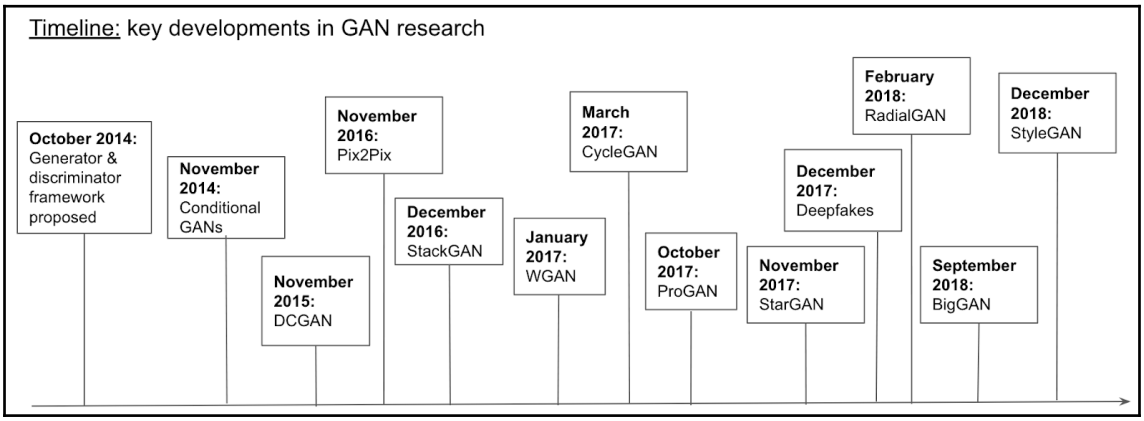

Discriminator
Real image
8
Fake image
8
Discriminator
Probability real
Probability fake

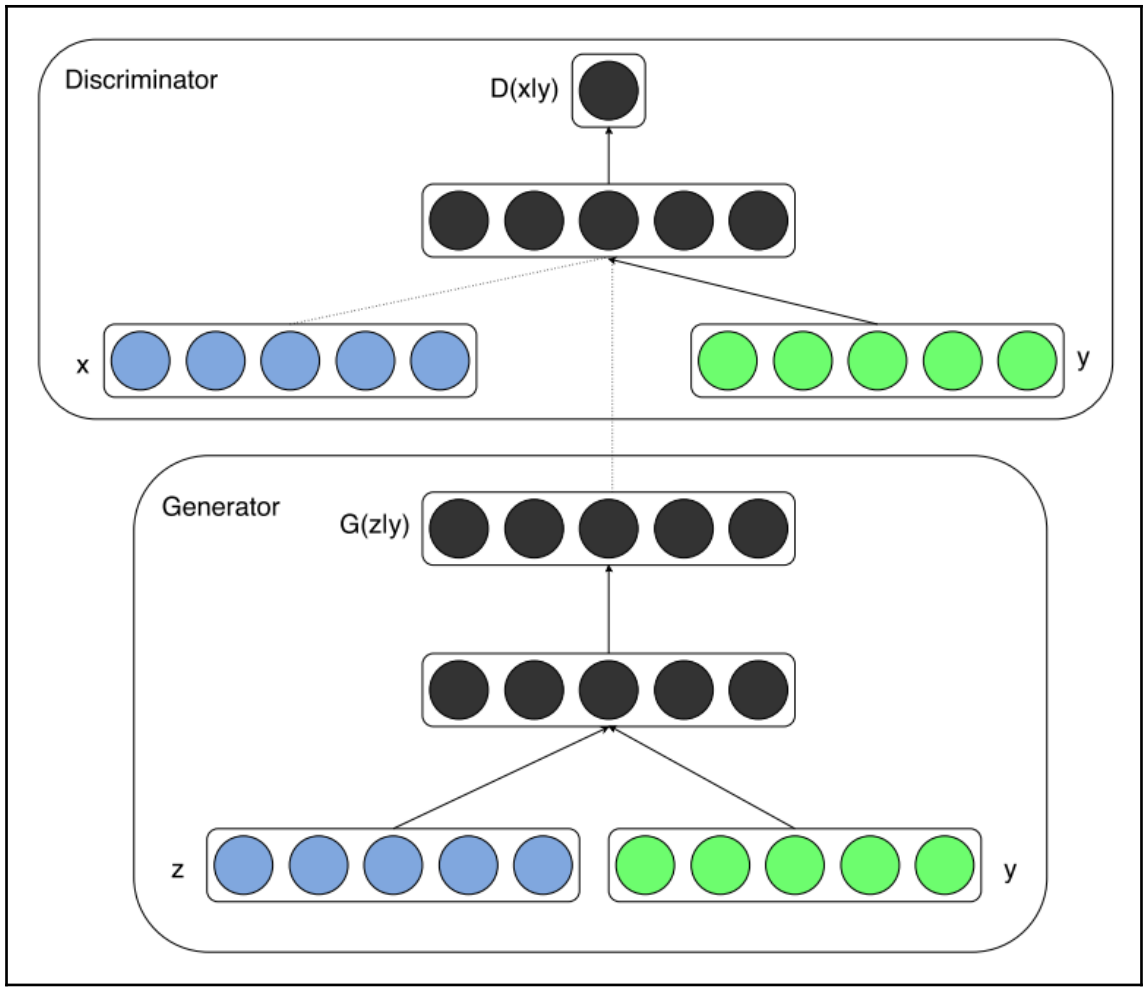

Generator
D-dimensional
noise vector
Fake image
8
Input to
discriminator
Upsampling

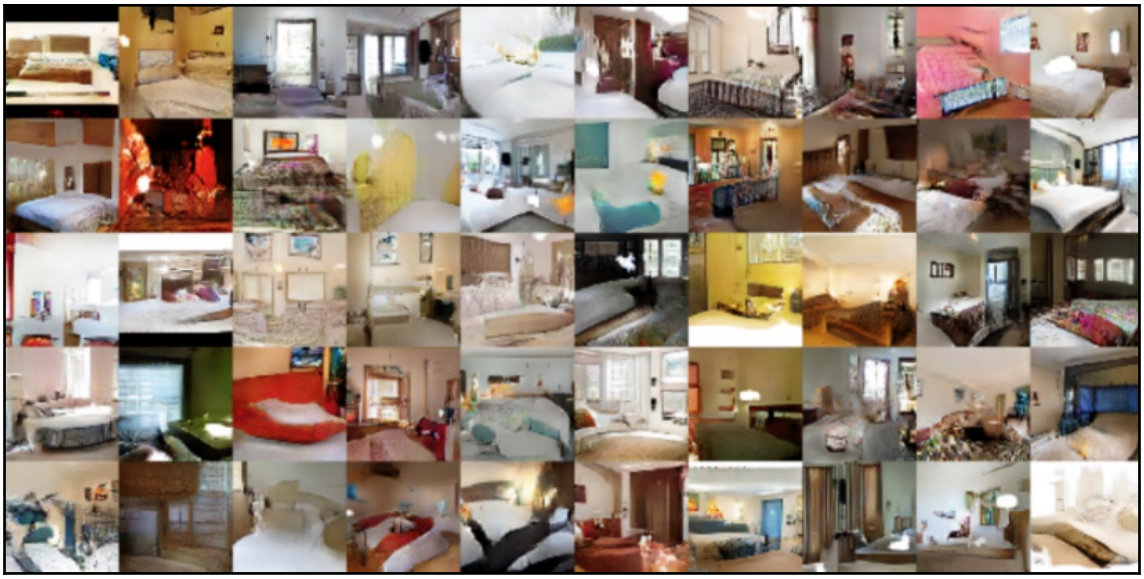

Logistic Sigmoid Function
f(x)
slopes near zero
x

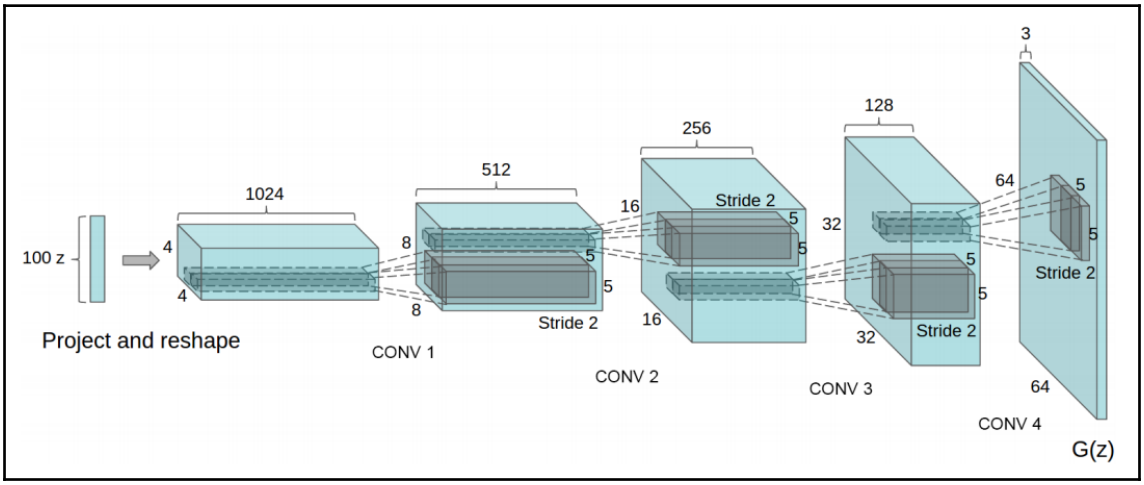

10k steps
20k steps
50K steps
100k steps

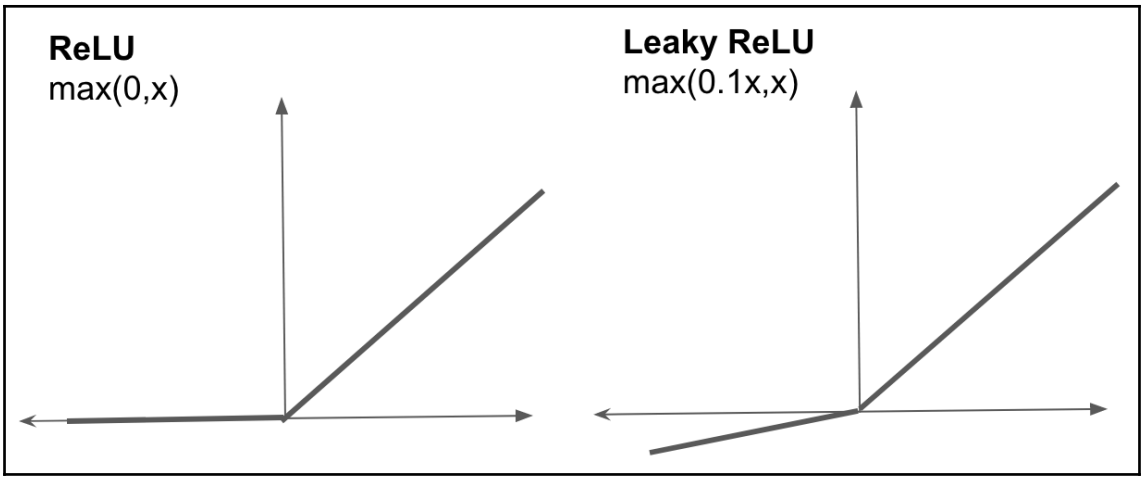

0.3


Timeline: key developments in GAN research
October 2014: Generator & discriminator framework proposed
November 2014: Conditional GANs
November 2015: DCGAN
November 2016: Pix2Pix
December 2016: StackGAN
January 2017: WGAN
March 2017: CycleGAN
October 2017: ProGAN
November 2017: StarGAN
December 2017: Deepfakes
February 2018: RadialGAN
September 2018: BigGAN
December 2018: StyleGAN


Discriminator
D(xly)
x
y
Generator
G(zly)
z
y



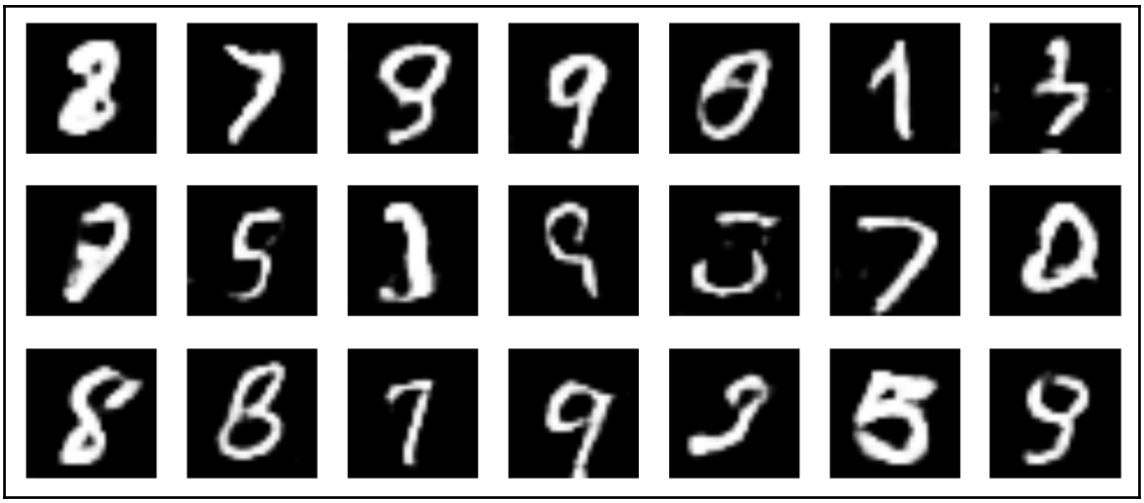

100 z
Project and reshape
1024
4
4
512
8
8
CONV 1
Stride 2
5
5
256
16
16
CONV 2
Stride 2
5
5
128
32
32
CONV 3
Stride 2
5
5
3
64
64
CONV 4
Stride 2
5
5
G(z)

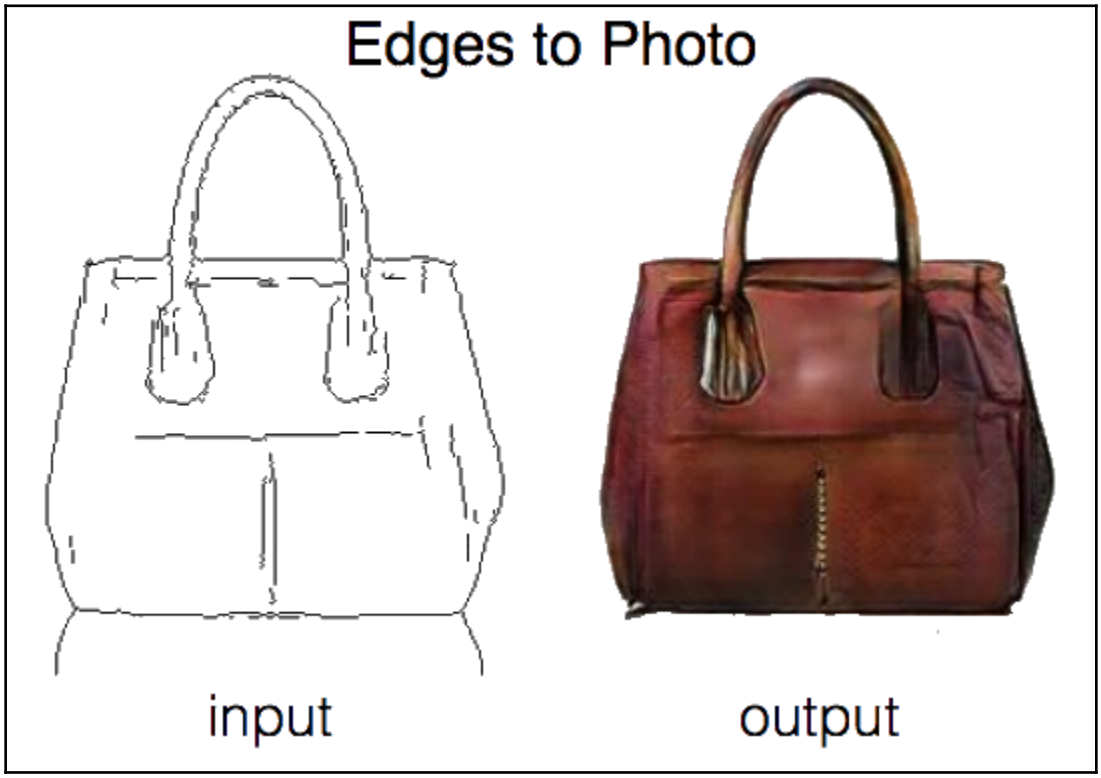


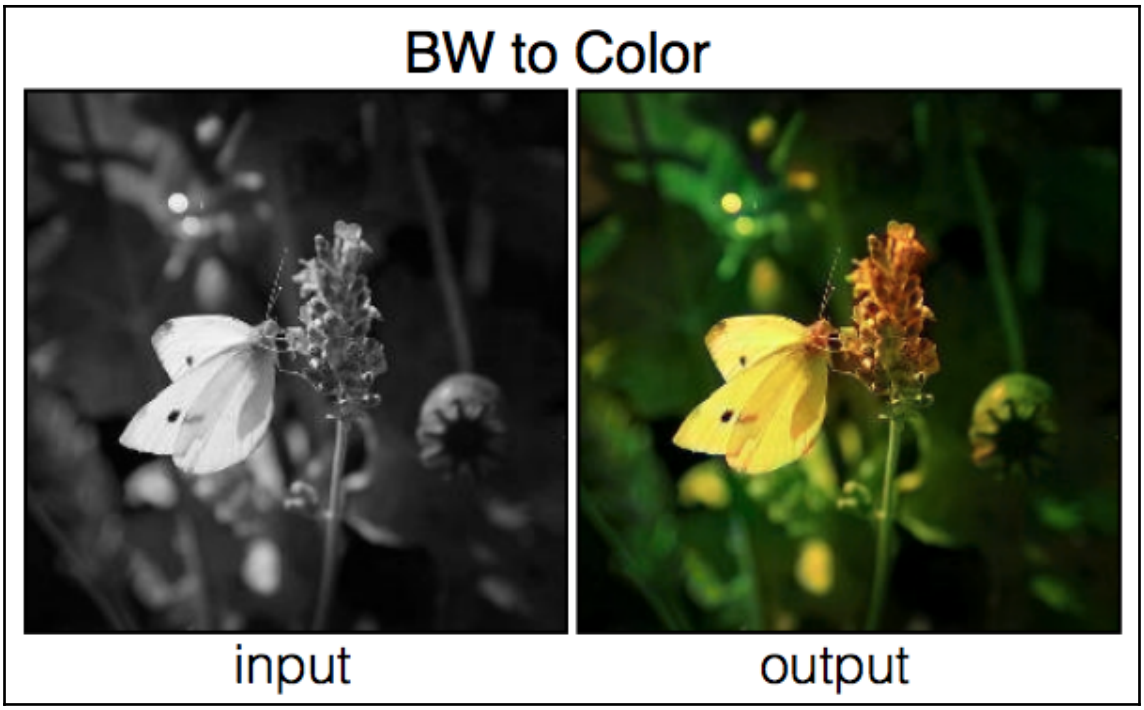

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_7 (Dense)              (None, 6272)              633472
_________________________________________________________________
batch_normalization_9 (Batch (None, 6272)              25088
_________________________________________________________________
reshape_5 (Reshape)          (None, 7, 7, 128)         0
_________________________________________________________________
up_sampling2d_9 (UpSampling2 (None, 14, 14, 128)       0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 14, 14, 64)        204864
_________________________________________________________________
batch_normalization_10 (Batc (None, 14, 14, 64)        256
_________________________________________________________________
up_sampling2d_10 (UpSampling (None, 28, 28, 64)        0
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 28, 28, 1)         1601
=================================================================
Total params: 865,281
Trainable params: 852,609
Non-trainable params: 12,672
_________________________________________________________________
None
```

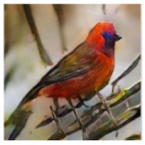
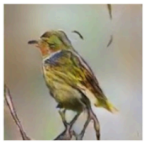
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 14, 14, 64)        1664
_________________________________________________________________
dropout_1 (Dropout)          (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 7, 7, 128)         204928
_________________________________________________________________
dropout_2 (Dropout)          (None, 7, 7, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 6272)              0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 6273
=================================================================
Total params: 212,865
Trainable params: 212,865
Non-trainable params: 0
_________________________________________________________________
```

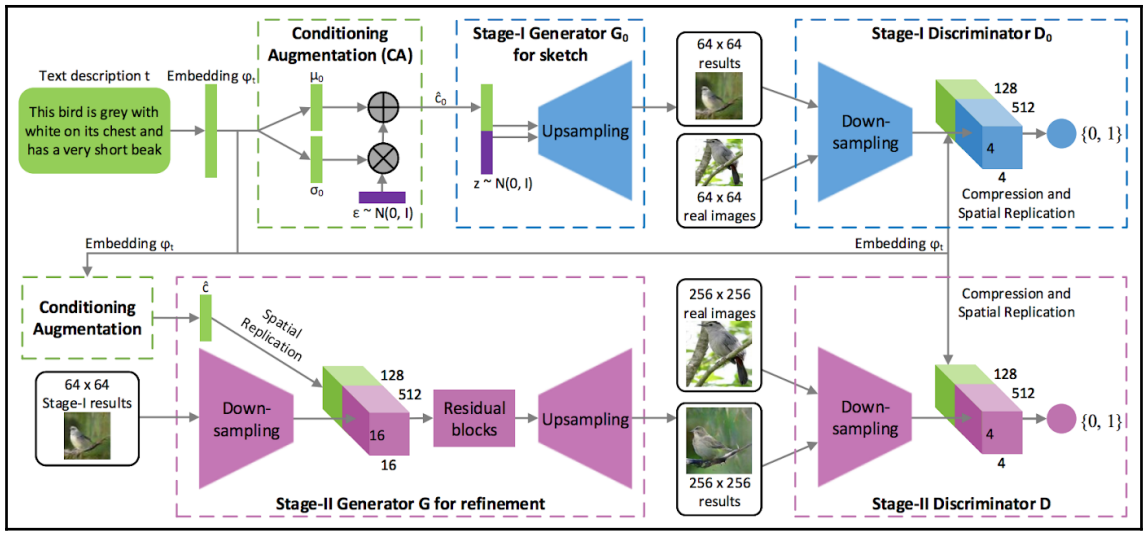
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         (None, 100)               0
_________________________________________________________________
sequential_1 (Sequential)    (None, 28, 28, 1)         865281
_________________________________________________________________
sequential_2 (Sequential)    (None, 1)                 212865
=================================================================
Total params: 1,078,146
Trainable params: 852,609
Non-trainable params: 225,537
_________________________________________________________________
```

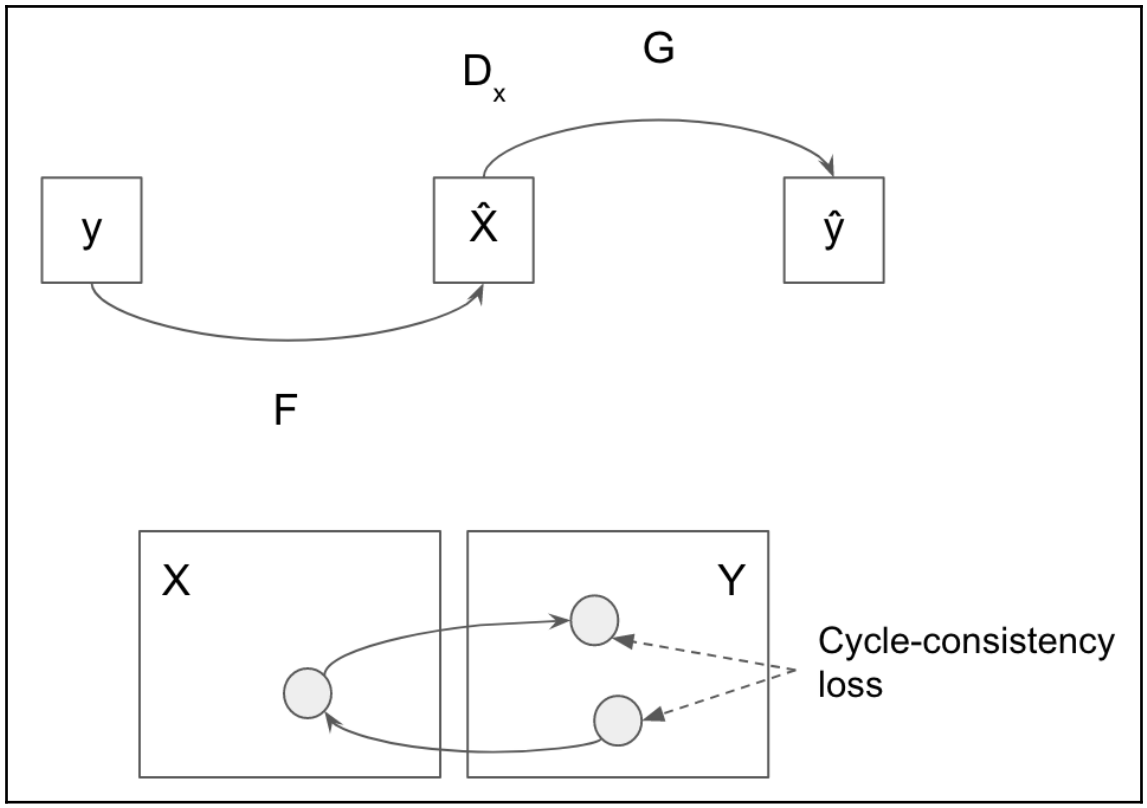


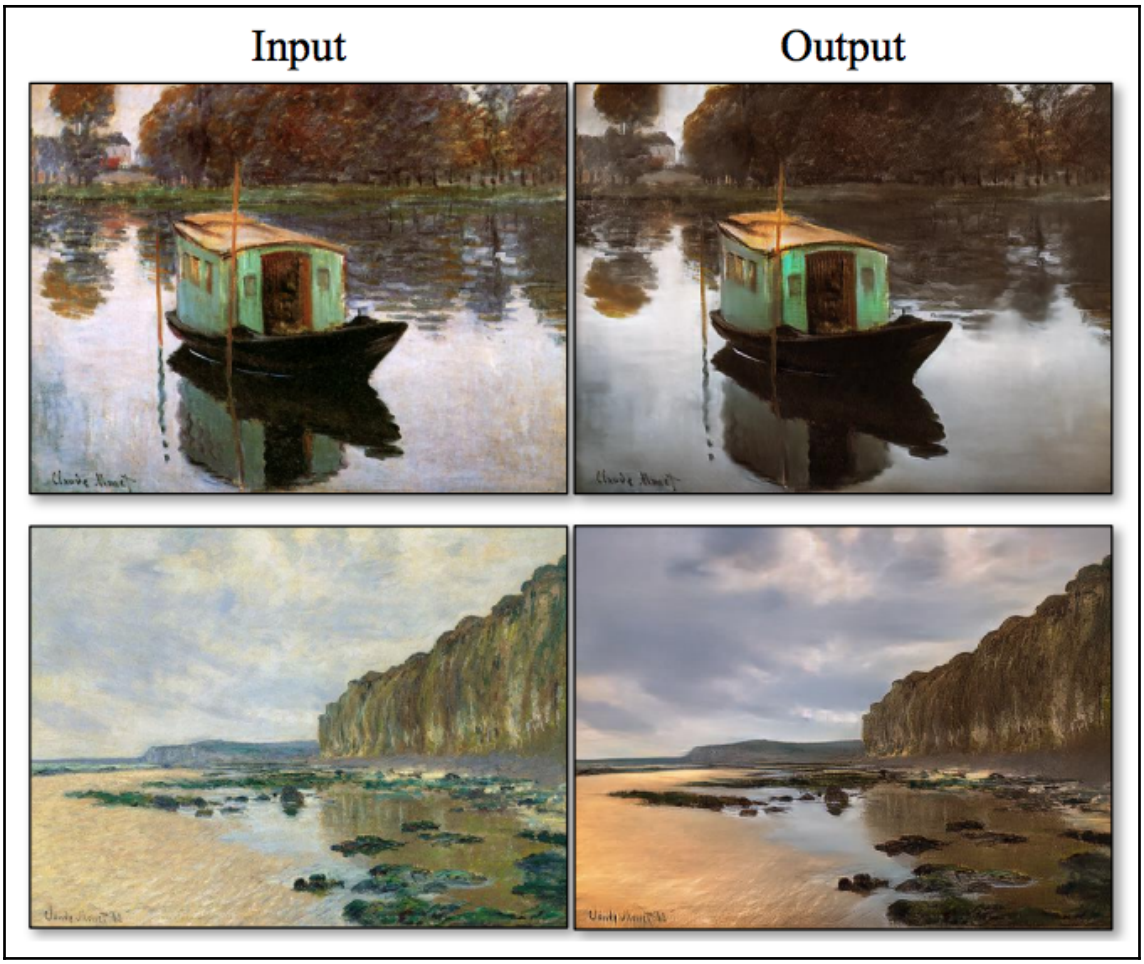

Edges to Photo
input
output

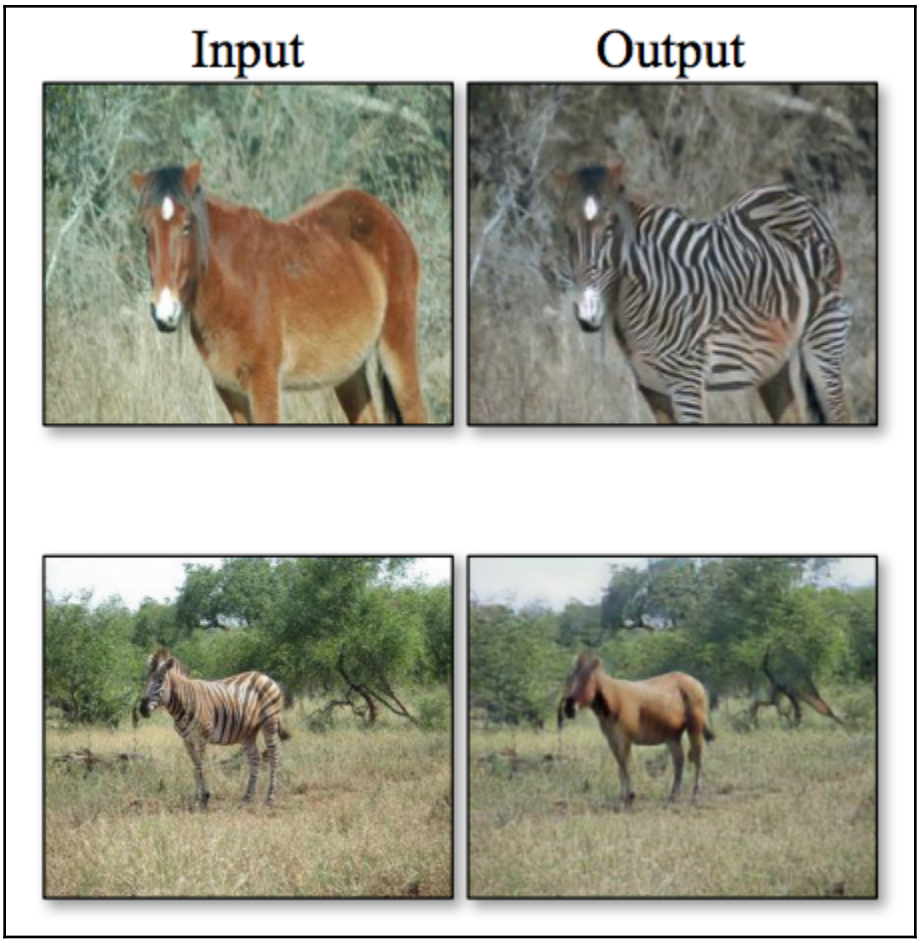


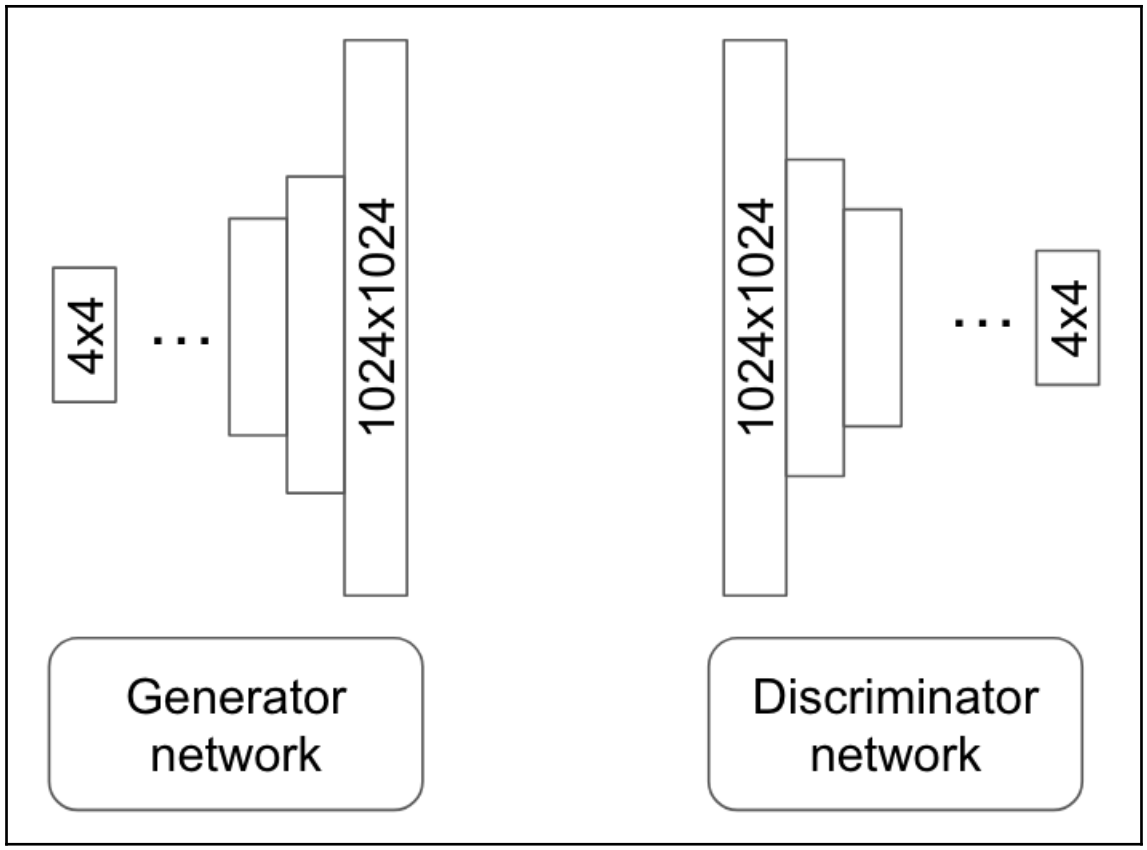
Text description	This bird is red and brown in color, with a stubby beak	The bird is short and stubby with yellow on its body	A bird with a medium orange bill white body gray wings and webbed feet	This small black bird has a short, slightly curved bill and long legs	A small bird with varying shades of brown with white under the eyes	A small yellow bird with a black crown and a short black pointed beak	This small bird has a white breast, light grey head, and black wings and tail
256x256 StackGAN							

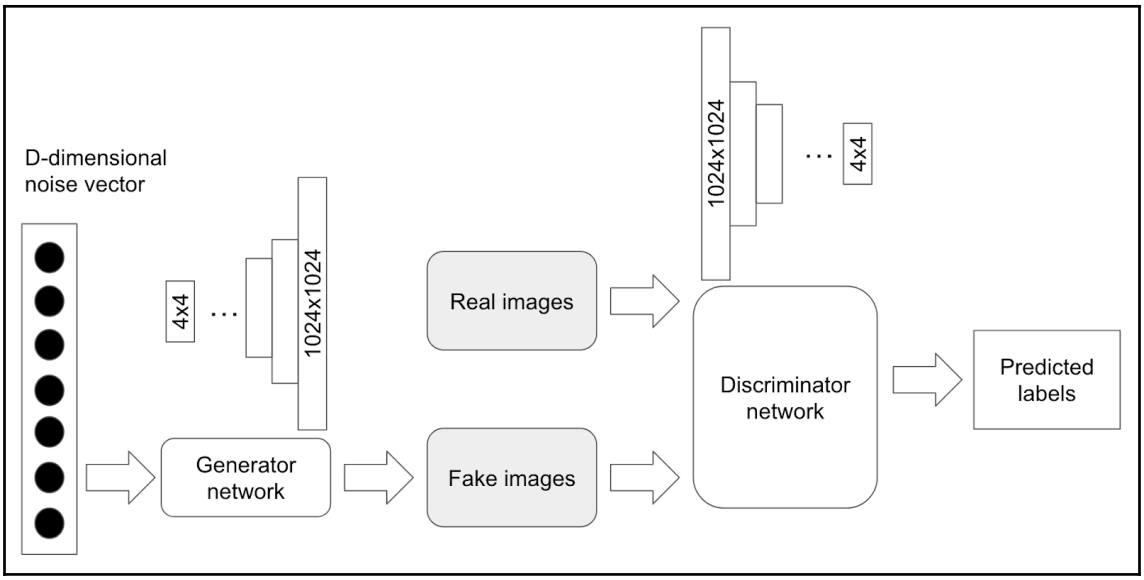

Text description t
This bird is grey with white on its chest and has a very short beak
Embedding φ_t
Conditioning Augmentation (CA)
μ_0
σ_0
$\varepsilon \sim N(0, I)$
$\hat{c}_0$
Stage-I Generator G_0 for sketch
$z \sim N(0, I)$
Upsampling
64 x 64 results
64 x 64 real images
Stage-I Discriminator D_0
Down-sampling
128
512
4
4
{0, 1}
Compression and Spatial Replication
Embedding φ_t
Embedding φ_t
Conditioning Augmentation
$\hat{c}$
Spatial Replication
64 x 64 Stage-I results
Down-sampling
128
512
16
16
Residual blocks
Upsampling
Stage-II Generator G for refinement
256 x 256 real images
256 x 256 results
Compression and Spatial Replication
Down-sampling
128
512
4
4
{0, 1}
Stage-II Discriminator D

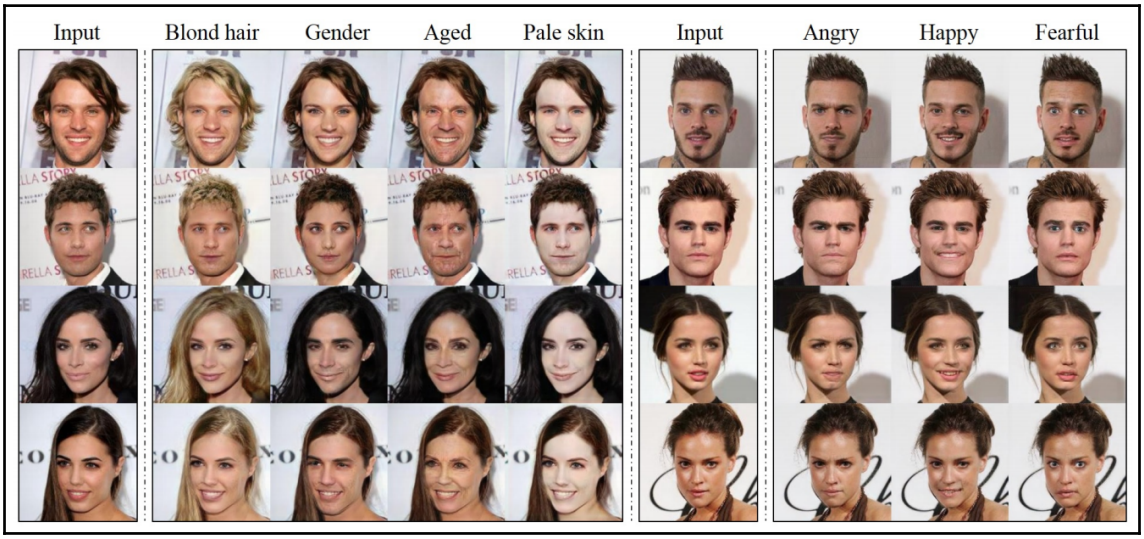

G
D_x
y
$\hat{X}$
$\hat{y}$
F
X
Y
Cycle-consistency loss

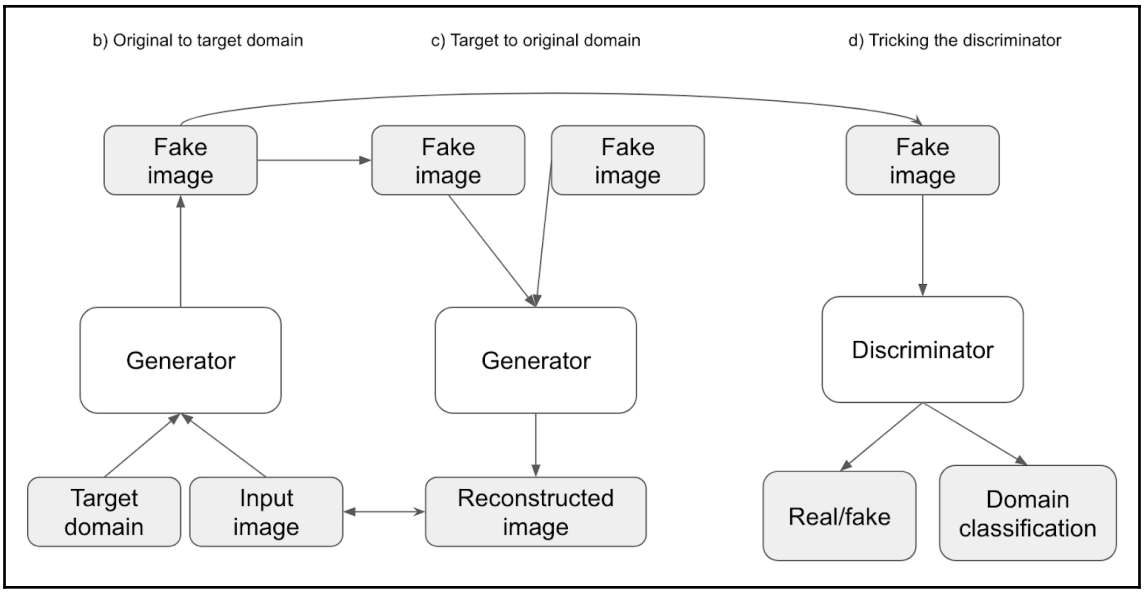

Input
Output


Input
Output

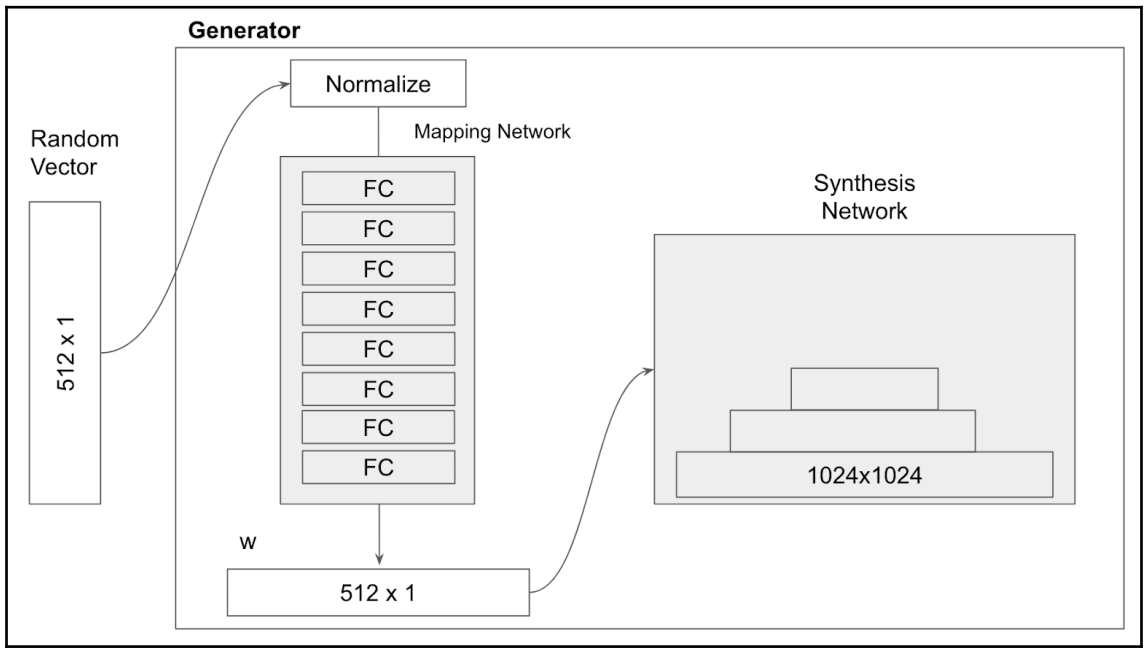

4x4
...
1024x1024
Generator network
1024x1024
...
4x4
Discriminator network

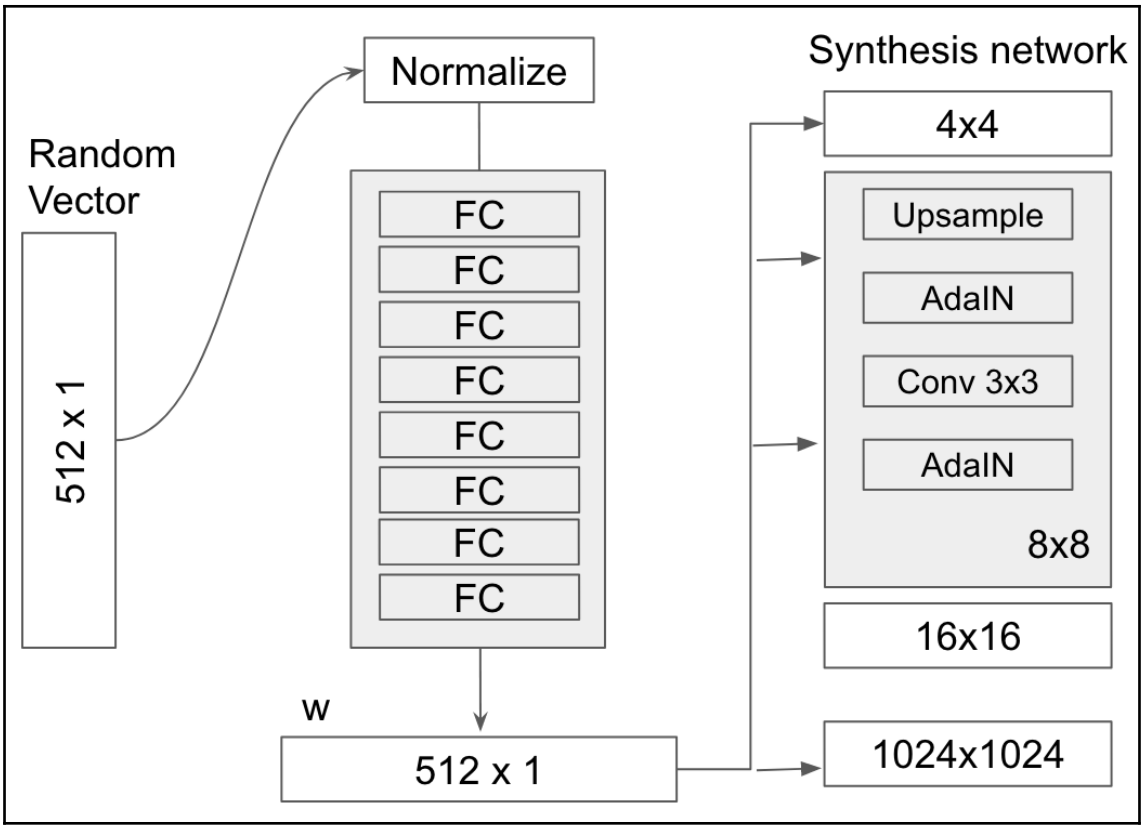

D-dimensional
noise vector
4x4
...
1024x1024
Generator
network
Real images
Fake images
1024x1024
...
4x4
Discriminator
network
Predicted
labels

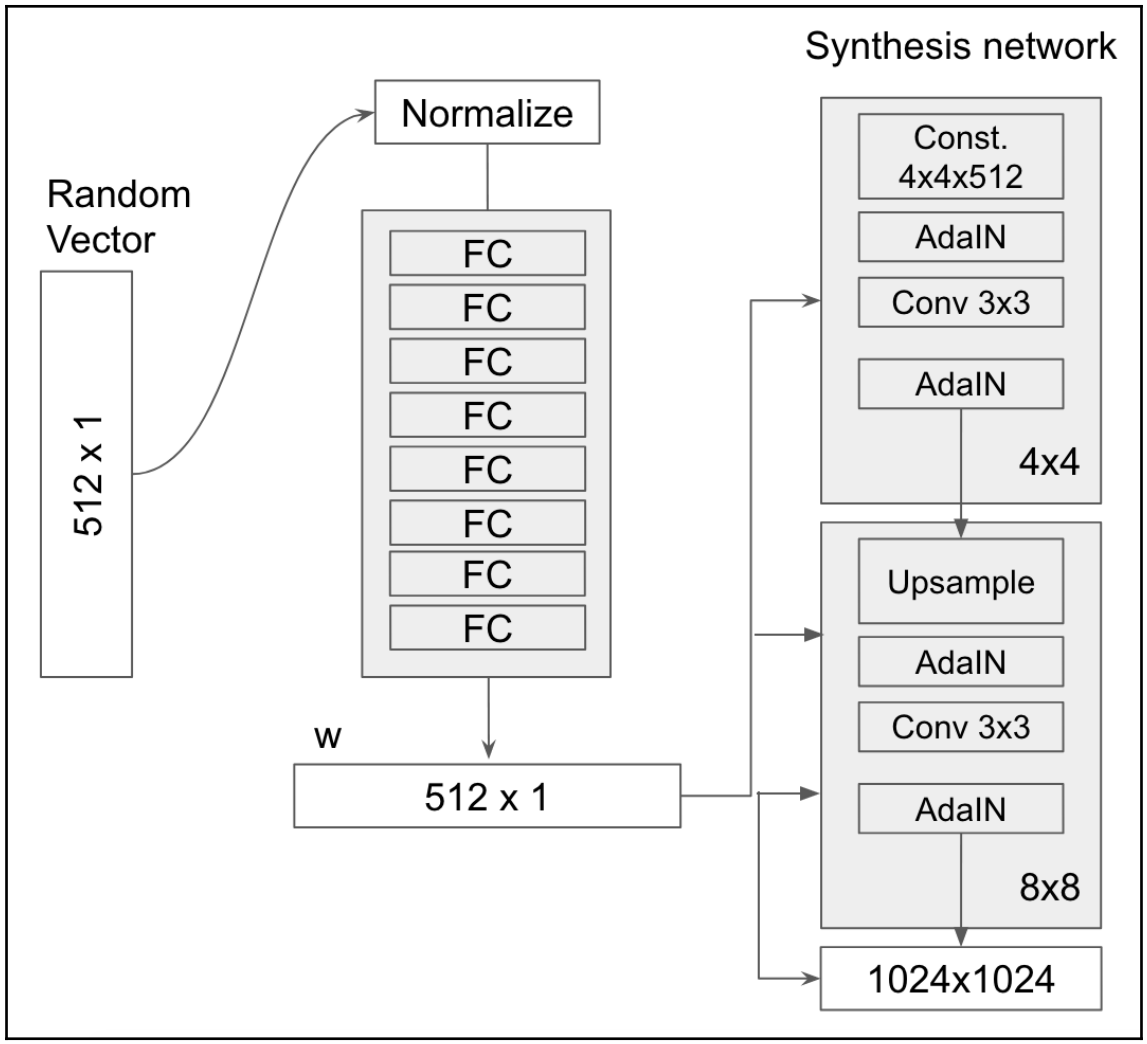

Input
Blond hair
Gender
Aged
Pale skin
Input
Angry
Happy
Fearful

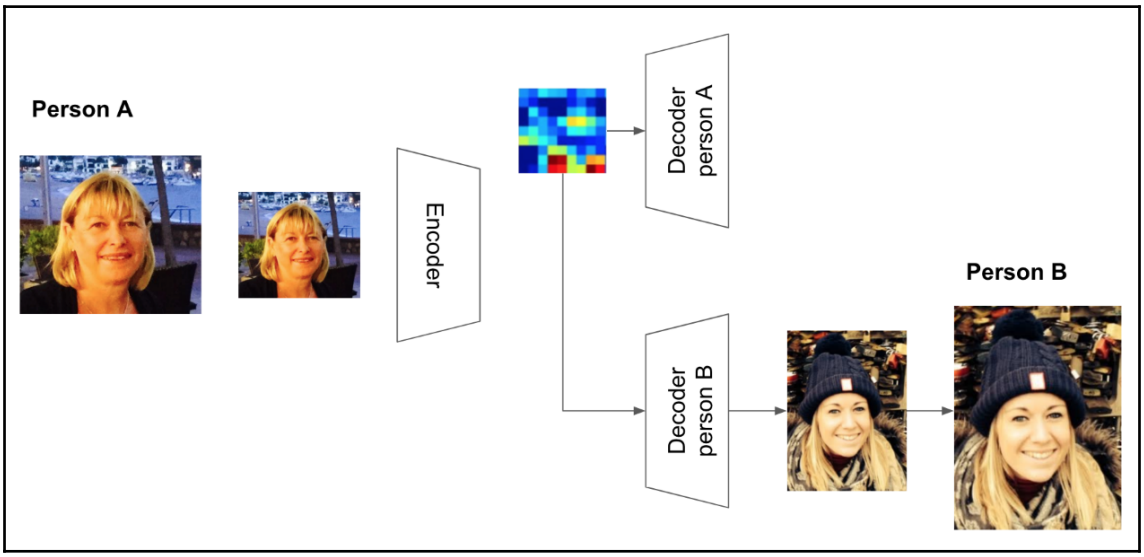

b) Original to target domain
c) Target to original domain
d) Tricking the discriminator
Fake image
Fake image
Fake image
Fake image
Generator
Generator
Discriminator
Target domain
Input image
Reconstructed image
Real/fake
Domain classification

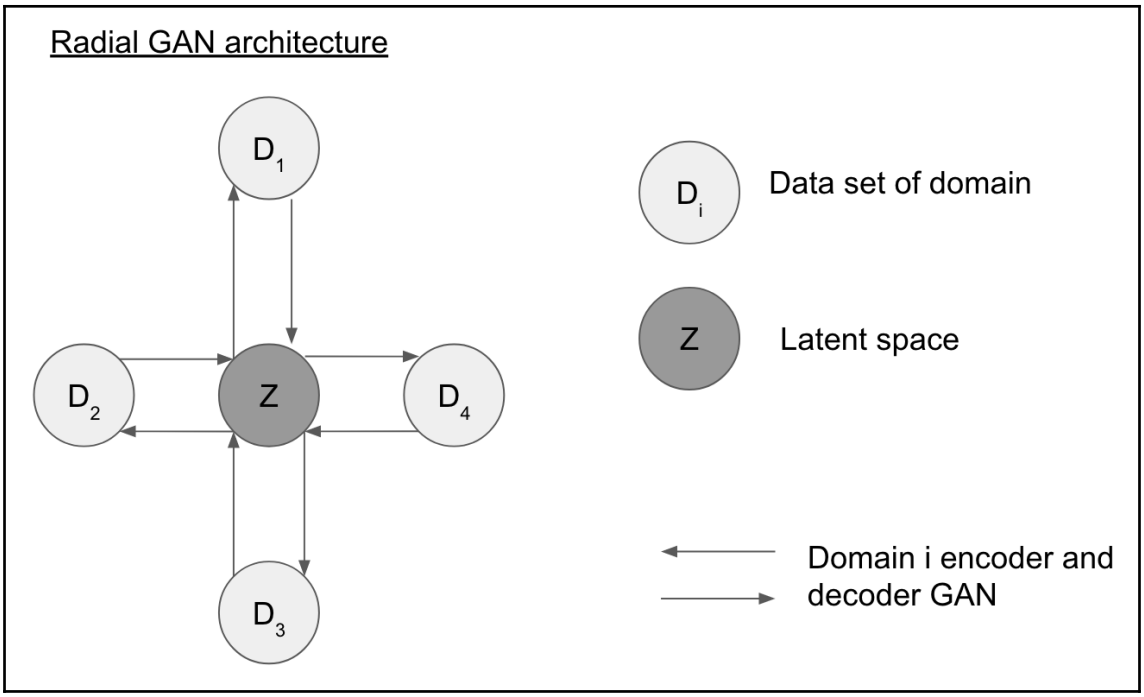

source
destination
Coarse styles copied


Generator
Normalize
Mapping Network
Random Vector
512 x 1
FC
FC
FC
FC
FC
FC
FC
FC
Synthesis Network
1024x1024
w
512 x 1

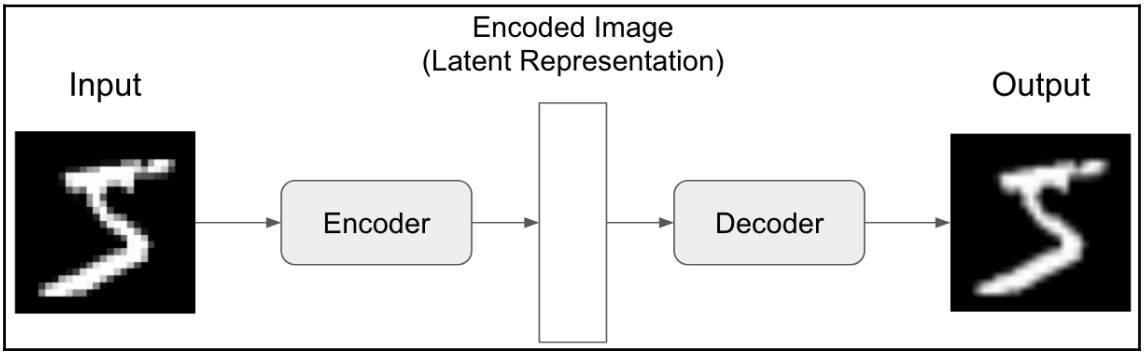

Normalize
Synthesis network
Random
Vector
512 x 1
FC
FC
FC
FC
FC
FC
FC
FC
4x4
Upsample
AdaIN
Conv 3x3
AdaIN
8x8
16x16
w
512 x 1
1024x1024

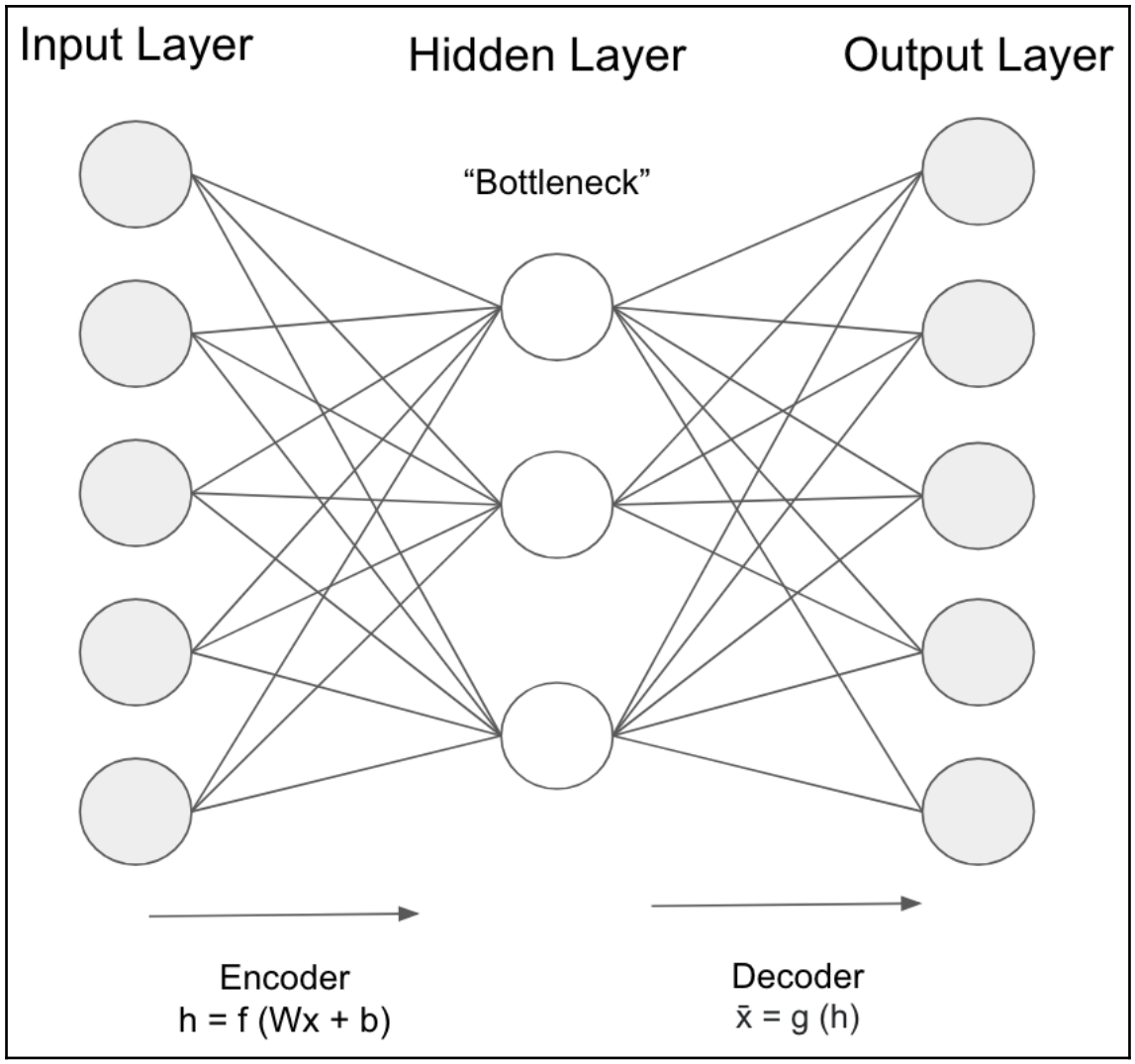

Synthesis network
Normalize
Random
Vector
512 x 1
FC
FC
FC
FC
FC
FC
FC
FC
w
512 x 1
Const.
4x4x512
AdaIN
Conv 3x3
AdaIN
4x4
Upsample
AdaIN
Conv 3x3
AdaIN
8x8
1024x1024


Person A
Encoder
Decoder person A
Decoder person B
Person B


Radial GAN architecture
D1
D2
Z
D4
D3
Di
Data set of domain
Z
Latent space
Domain i encoder and
decoder GAN

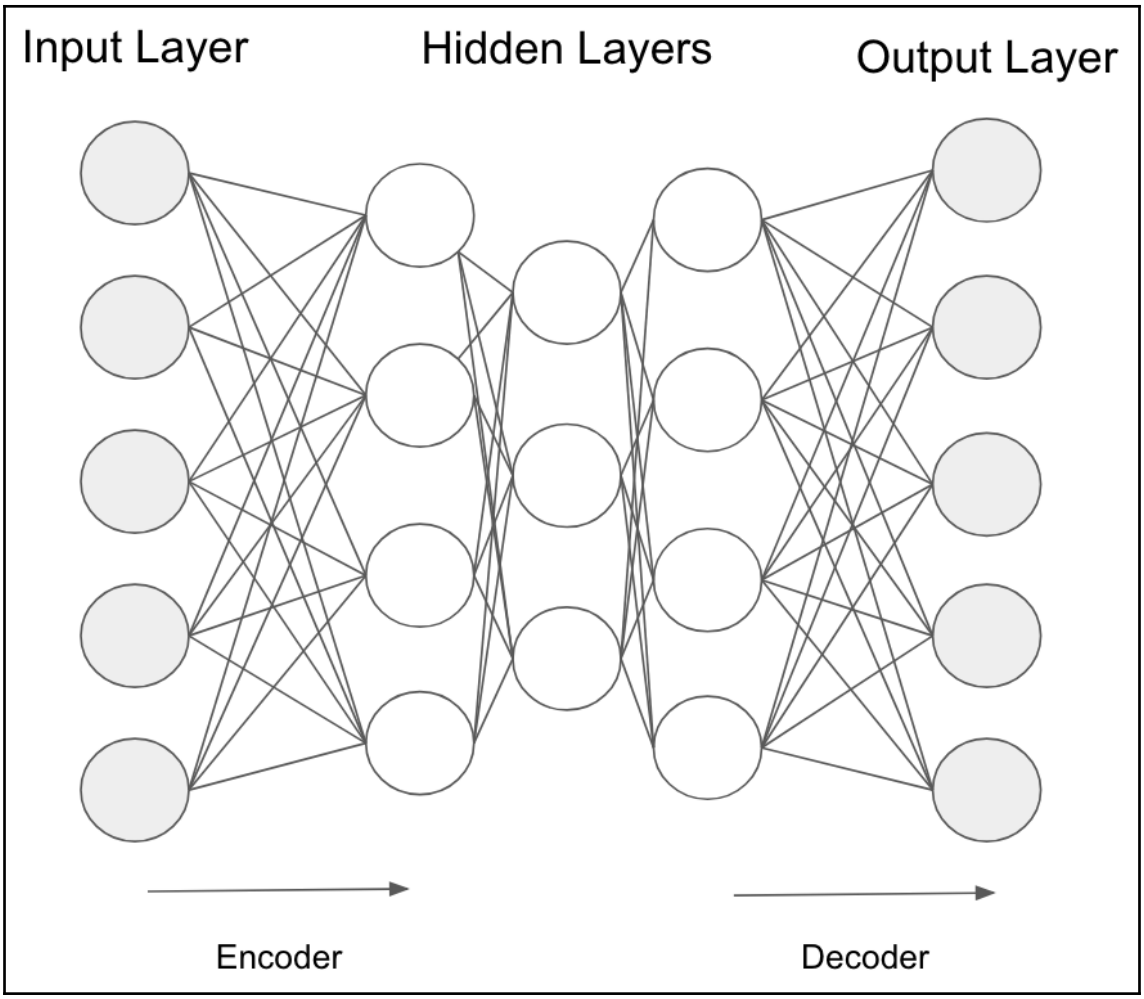
Chapter 8: Implementing Autoencoders




Input Layer
Hidden Layer
Output Layer
“Bottleneck”
Encoder
h = f (Wx + b)
Decoder
x̄ = g (h)

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_4 (InputLayer)         (None, 784)               0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                50240
_________________________________________________________________
dense_2 (Dense)              (None, 784)               50960
=================================================================
Total params: 101,200
Trainable params: 101,200
Non-trainable params: 0
_________________________________________________________________
```

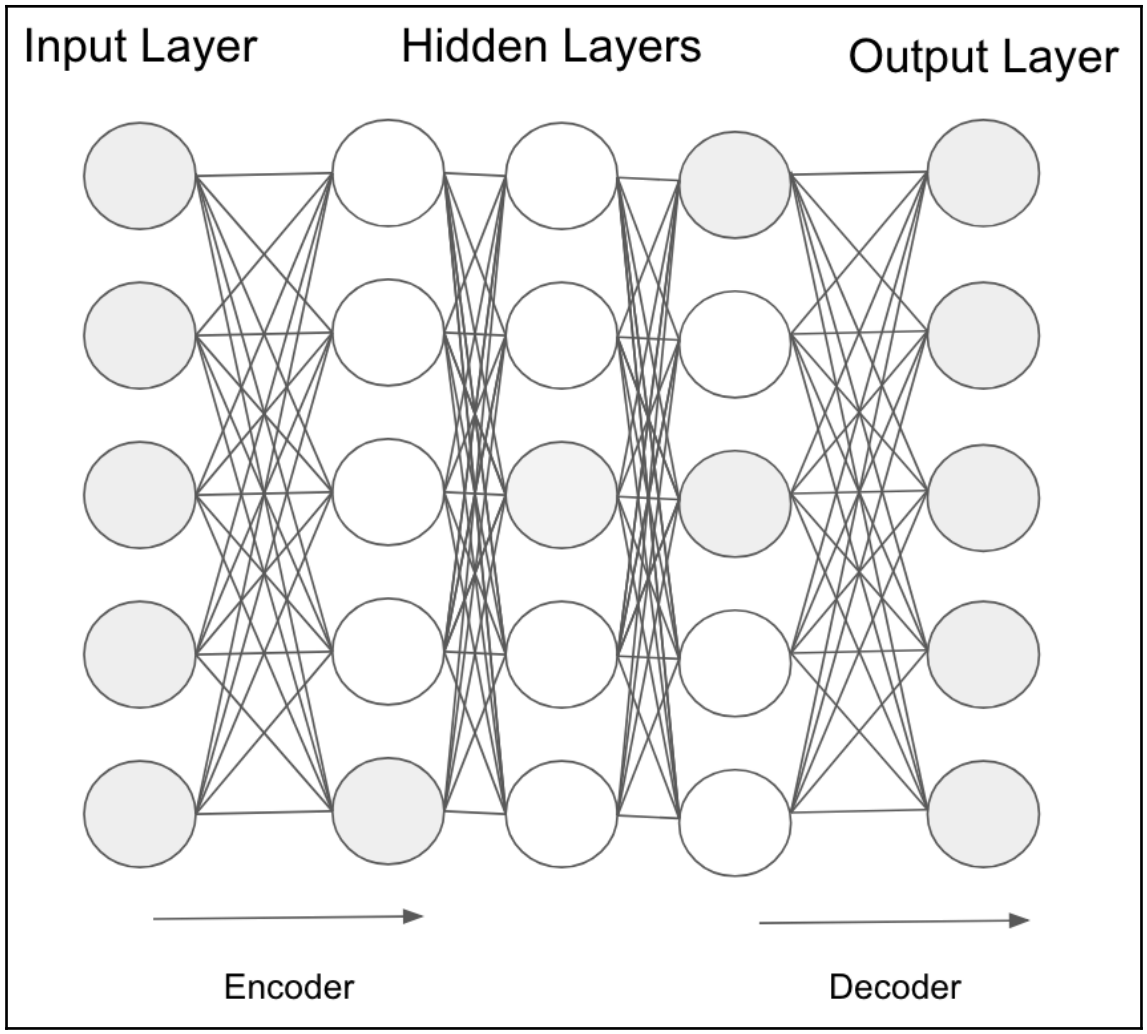
```
Train on 60000 samples, validate on 10000 samples
Epoch 1/10
60000/60000 [==============================] - 3s 42us/step - loss: 0.0446 - val_loss: 0.0221
Epoch 2/10
60000/60000 [==============================] - 1s 22us/step - loss: 0.0172 - val_loss: 0.0128
Epoch 3/10
60000/60000 [==============================] - 1s 22us/step - loss: 0.0109 - val_loss: 0.0087
Epoch 4/10
60000/60000 [==============================] - 1s 22us/step - loss: 0.0077 - val_loss: 0.0065
Epoch 5/10
60000/60000 [==============================] - 1s 22us/step - loss: 0.0061 - val_loss: 0.0055
Epoch 6/10
60000/60000 [==============================] - 1s 22us/step - loss: 0.0053 - val_loss: 0.0049
Epoch 7/10
60000/60000 [==============================] - 1s 22us/step - loss: 0.0048 - val_loss: 0.0045
Epoch 8/10
60000/60000 [==============================] - 1s 22us/step - loss: 0.0046 - val_loss: 0.0043
Epoch 9/10
60000/60000 [==============================] - 1s 22us/step - loss: 0.0044 - val_loss: 0.0042
Epoch 10/10
60000/60000 [==============================] - 1s 22us/step - loss: 0.0043 - val_loss: 0.0041
```

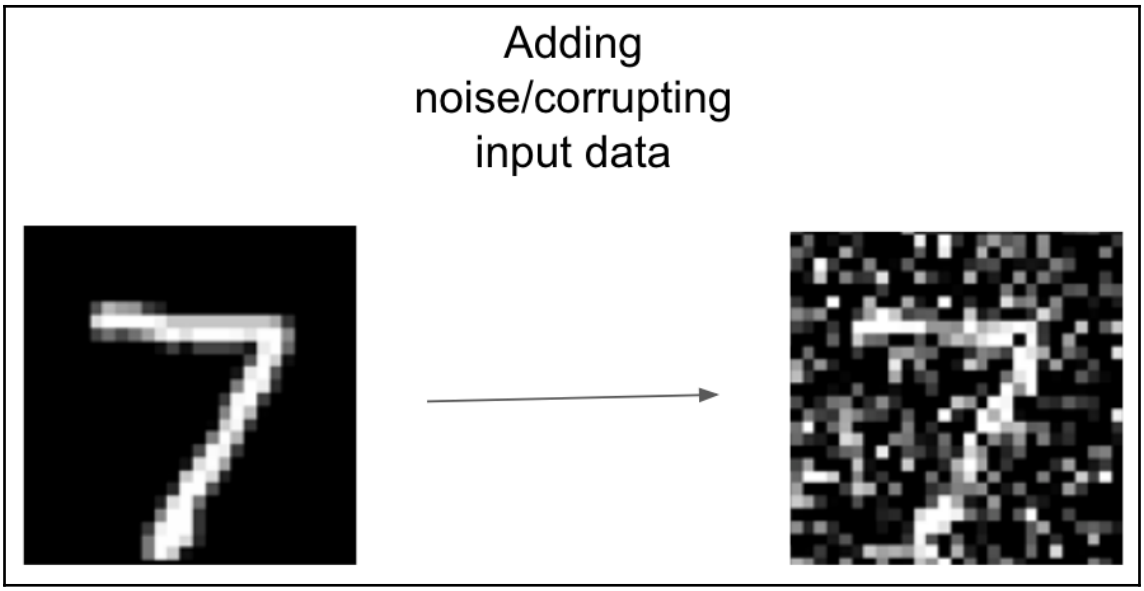

Input Layer
Hidden Layers
Output Layer
Encoder
Decoder

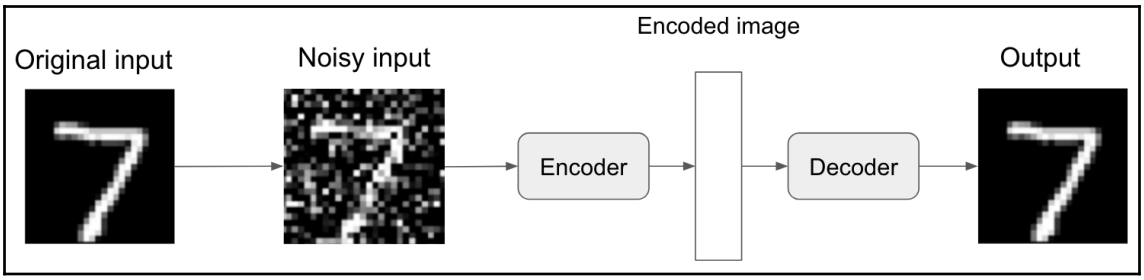
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         (None, 784)               0
_________________________________________________________________
dense_7 (Dense)              (None, 128)               100480
_________________________________________________________________
dense_8 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_9 (Dense)              (None, 128)               8320
_________________________________________________________________
dense_10 (Dense)             (None, 784)               101136
=================================================================
Total params: 218,192
Trainable params: 218,192
Non-trainable params: 0
_________________________________________________________________
```

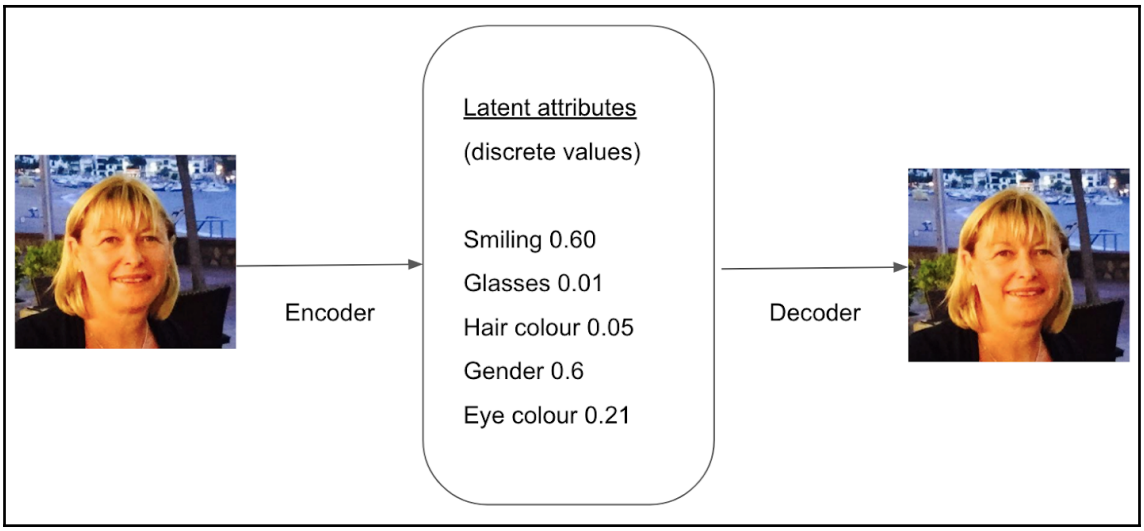
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)            (None, 28, 28, 32)        320
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 14, 14, 64)        18496
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 7, 7, 64)          0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 4, 4, 128)         73856
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 4, 4, 128)         147584
_________________________________________________________________
up_sampling2d_1 (UpSampling2 (None, 8, 8, 128)         0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 8, 8, 8)           4104
_________________________________________________________________
up_sampling2d_2 (UpSampling2 (None, 16, 16, 8)         0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 14, 14, 64)        4672
_________________________________________________________________
up_sampling2d_3 (UpSampling2 (None, 28, 28, 64)        0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 28, 28, 1)         577
=================================================================
Total params: 249,609
Trainable params: 249,609
Non-trainable params: 0
```

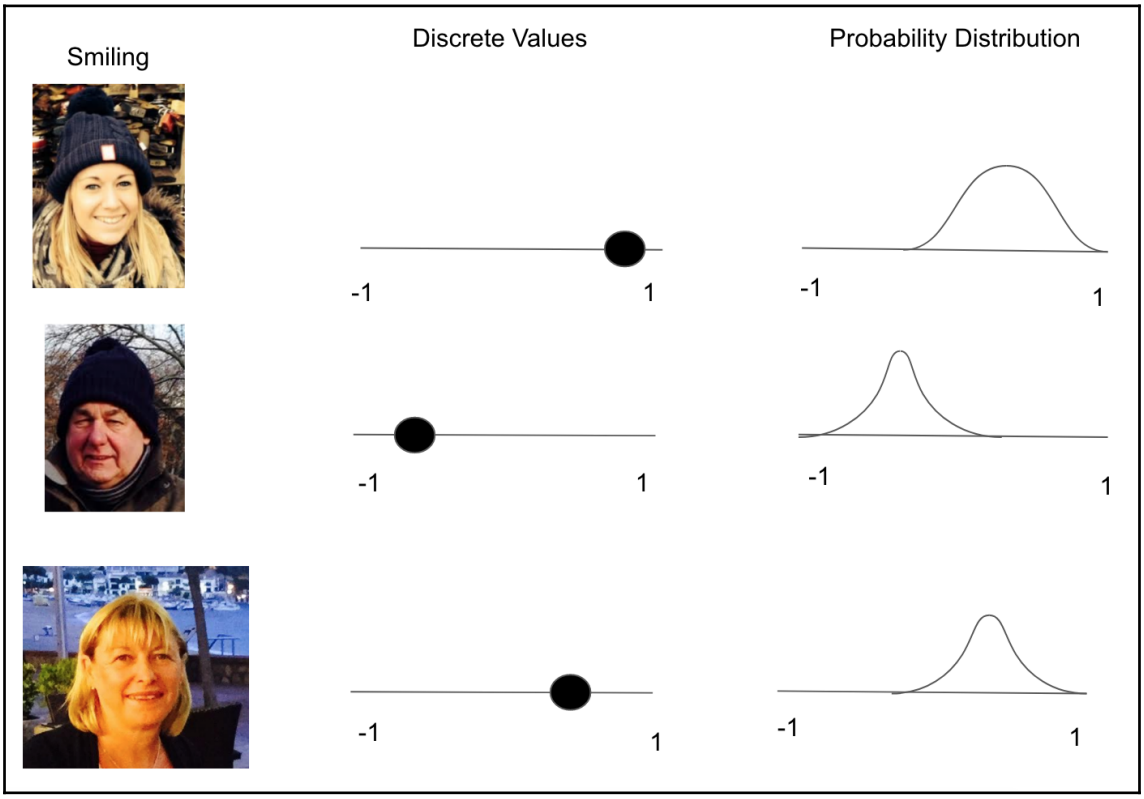
```
Train on 60000 samples, validate on 10000 samples
Epoch 1/10
60000/60000 [==============================] - 185s 3ms/step - loss: 0.1039 - val_loss: 0.1011
Epoch 2/10
60000/60000 [==============================] - 188s 3ms/step - loss: 0.1012 - val_loss: 0.0993
Epoch 3/10
60000/60000 [==============================] - 166s 3ms/step - loss: 0.0992 - val_loss: 0.0970
Epoch 4/10
60000/60000 [==============================] - 173s 3ms/step - loss: 0.0975 - val_loss: 0.0956
Epoch 5/10
60000/60000 [==============================] - 180s 3ms/step - loss: 0.0963 - val_loss: 0.0951
Epoch 6/10
60000/60000 [==============================] - 211s 4ms/step - loss: 0.0951 - val_loss: 0.0936
Epoch 7/10
60000/60000 [==============================] - 158s 3ms/step - loss: 0.0942 - val_loss: 0.0927
Epoch 8/10
60000/60000 [==============================] - 166s 3ms/step - loss: 0.0933 - val_loss: 0.0919
Epoch 9/10
60000/60000 [==============================] - 176s 3ms/step - loss: 0.0925 - val_loss: 0.0912
Epoch 10/10
60000/60000 [==============================] - 171s 3ms/step - loss: 0.0918 - val_loss: 0.0906
```

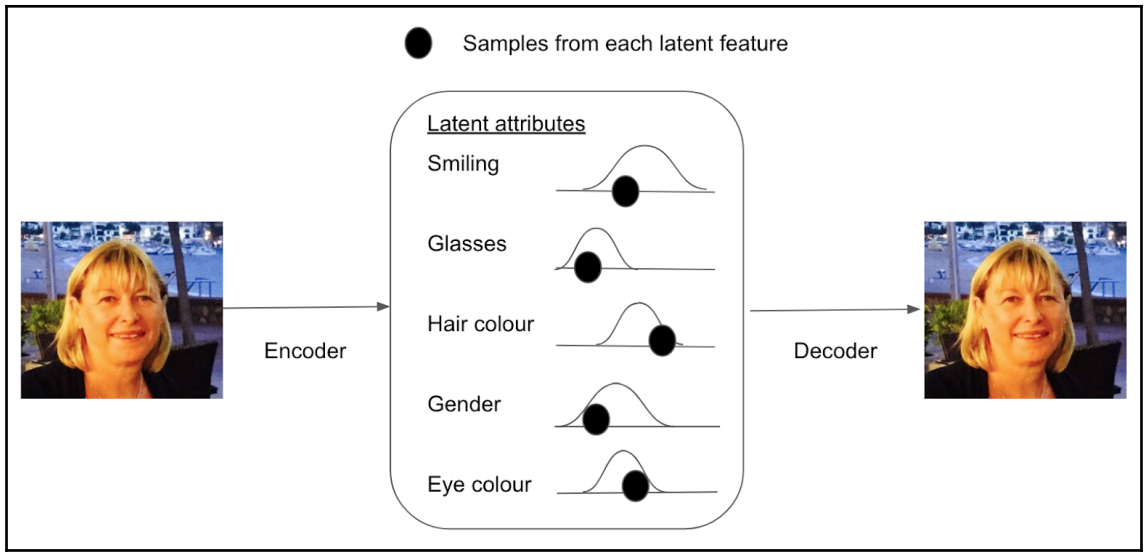

Input Layer
Hidden Layers
Output Layer
Encoder
Decoder

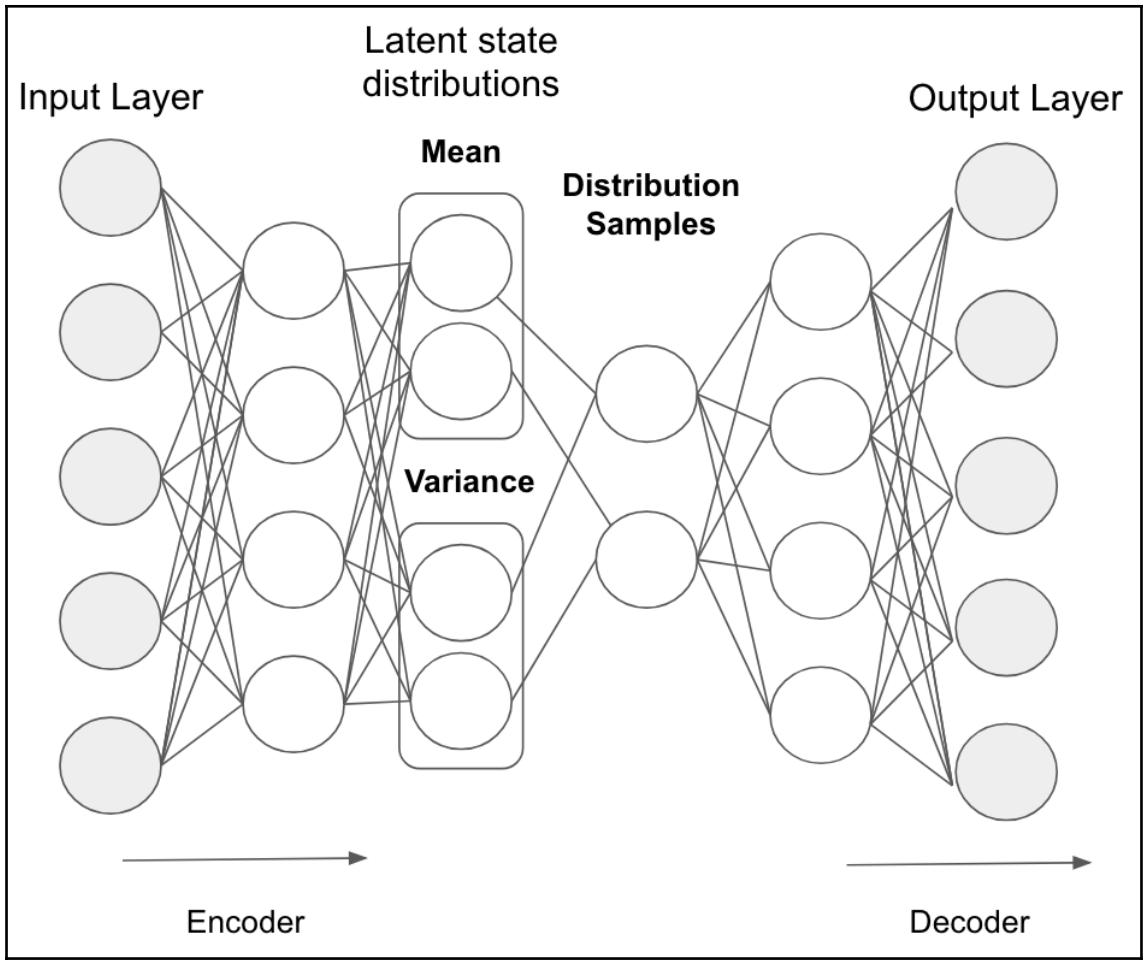

Adding
noise/corrupting
input data

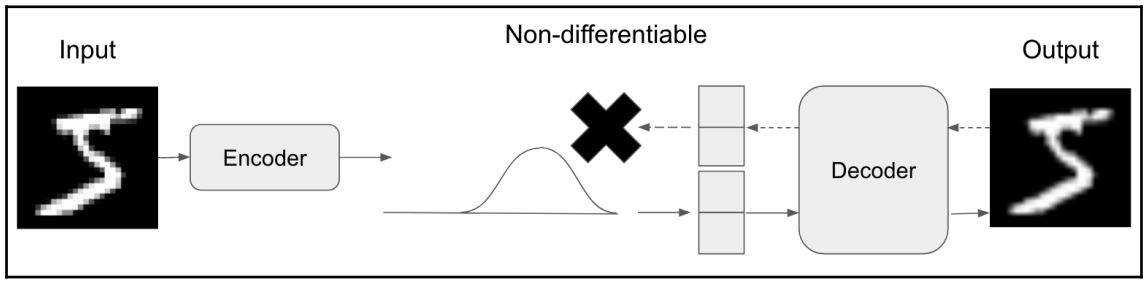

Original input
Noisy input
Encoded image
Encoder
Decoder
Output

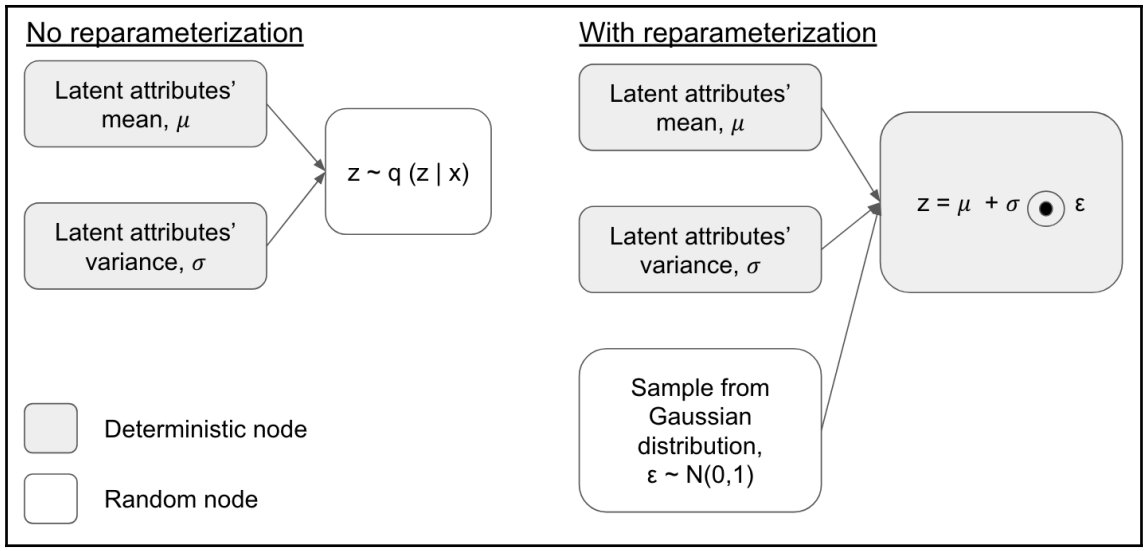

Latent attributes
(discrete values)
Smiling 0.60
Glasses 0.01
Hair colour 0.05
Gender 0.6
Eye colour 0.21
Encoder
Decoder

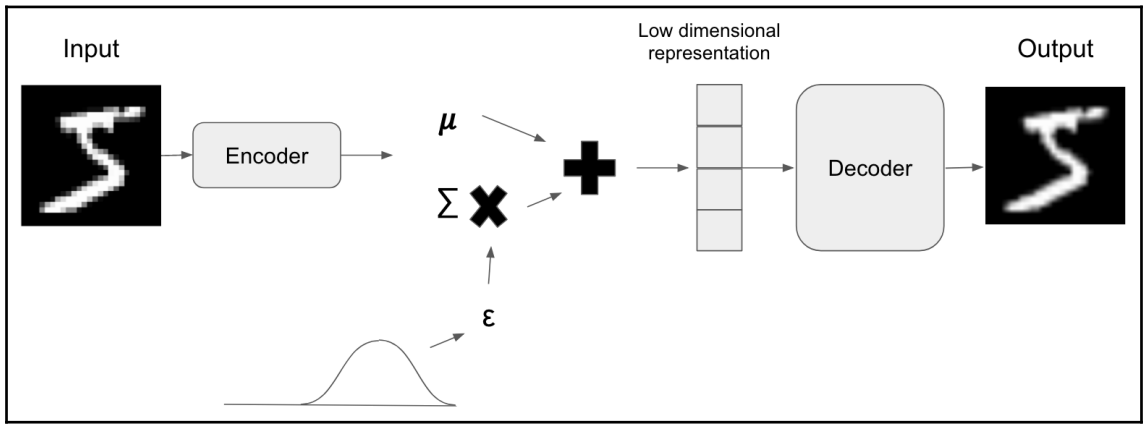

Smiling
Discrete Values
Probability Distribution
-1
1
-1
1
-1
1
-1
1
-1
1
-1
1


Samples from each latent feature
Latent attributes
Smiling
Glasses
Hair colour
Gender
Eye colour
Encoder
Decoder

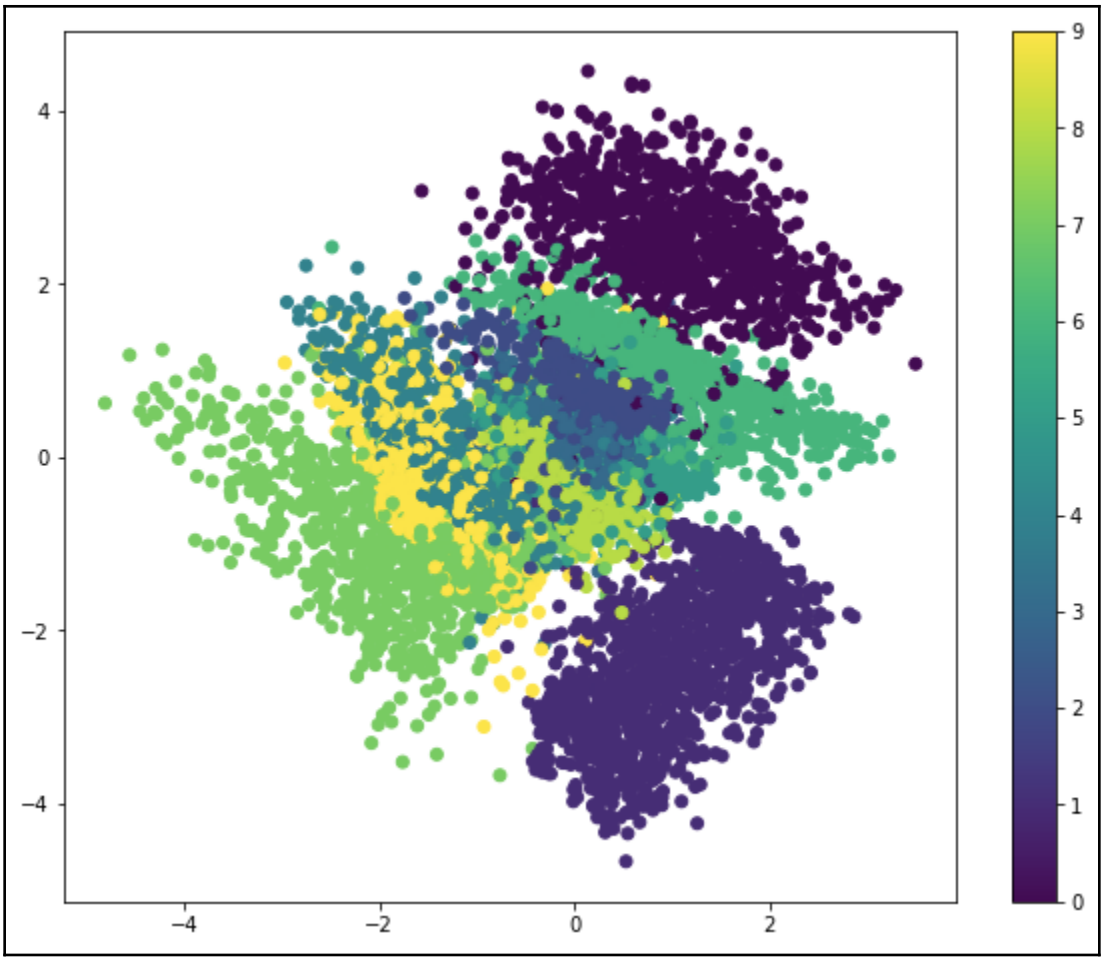

Latent state distributions
Input Layer
Output Layer
Mean
Distribution Samples
Variance
Encoder
Decoder

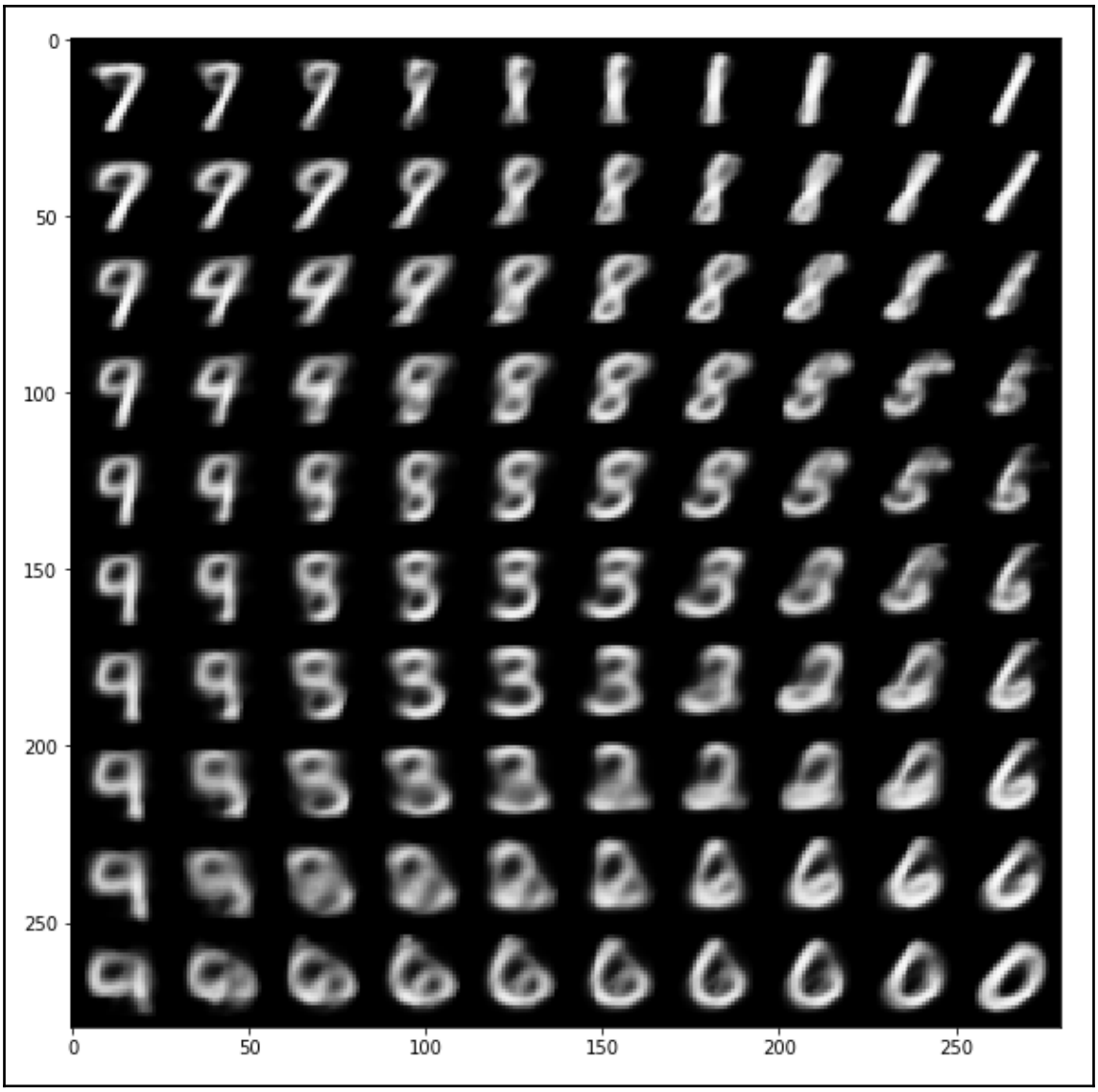

Input
Non-differentiable
Output
Encoder
Decoder


No reparameterization
Latent attributes' mean, μ
Latent attributes' variance, σ
z ~ q (z | x)
Deterministic node
Random node
With reparameterization
Latent attributes' mean, μ
Latent attributes' variance, σ
Sample from Gaussian distribution, ε ~ N(0,1)
z = μ + σ ⊙ ε

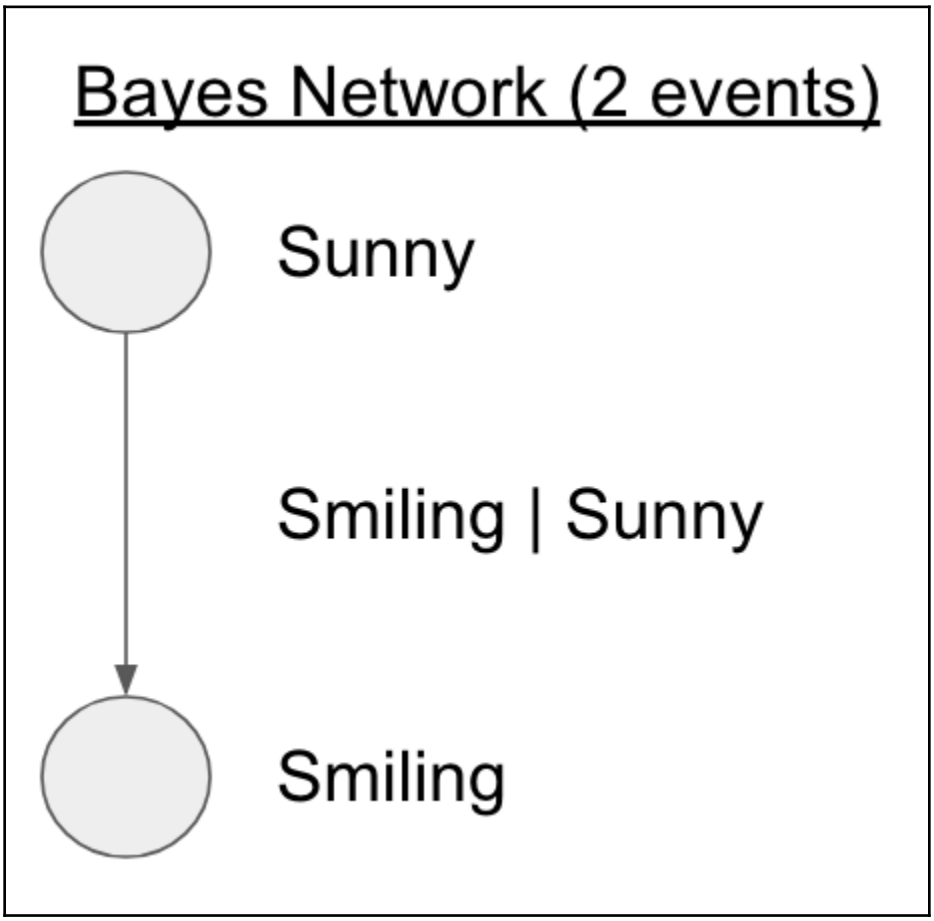

Input
Encoder
μ
Σ
ε
Low dimensional representation
Decoder
Output

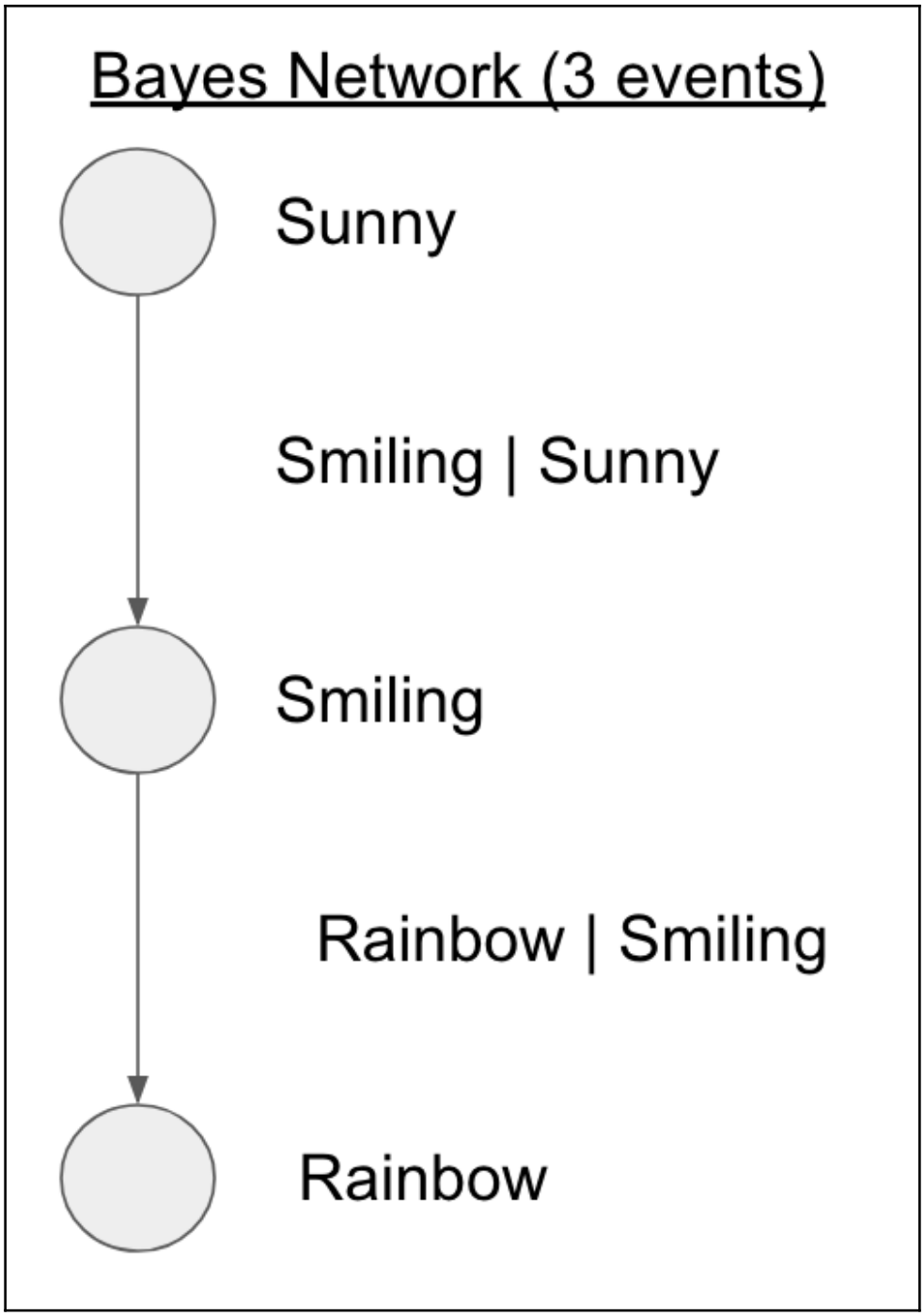
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   Connected to
=================================================================
input_1 (InputLayer)         (None, 784)               0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               200960    input_1[0][0]
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 514       dense_1[0][0]
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 514       dense_1[0][0]
_________________________________________________________________
lambda_1 (Lambda)            (None, 2)                 0         dense_2[0][0]
                                                                 dense_3[0][0]
_________________________________________________________________
dense_4 (Dense)              (None, 256)               768       lambda_1[0][0]
_________________________________________________________________
dense_5 (Dense)              (None, 784)               201488    dense_4[0][0]
=================================================================
Total params: 404,244
Trainable params: 404,244
Non-trainable params: 0
```

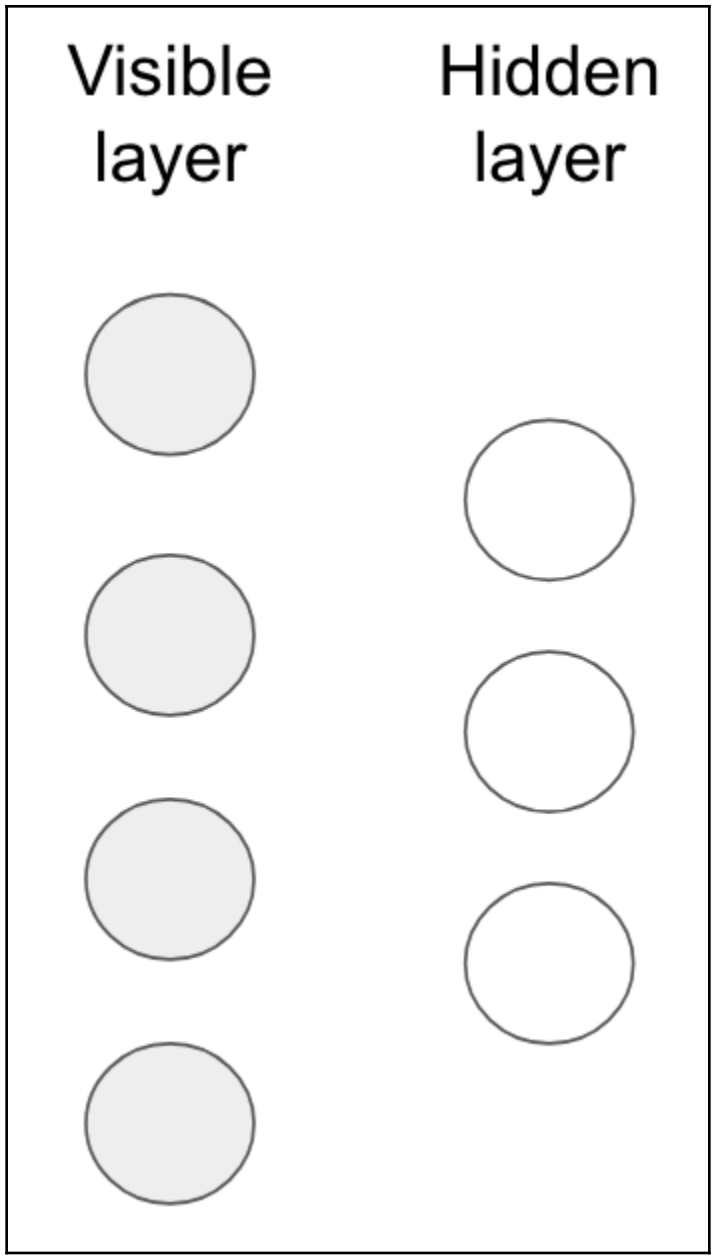

4
2
0
−2
−4
−4
−2
0
2
9
8
7
6
5
4
3
2
1
0

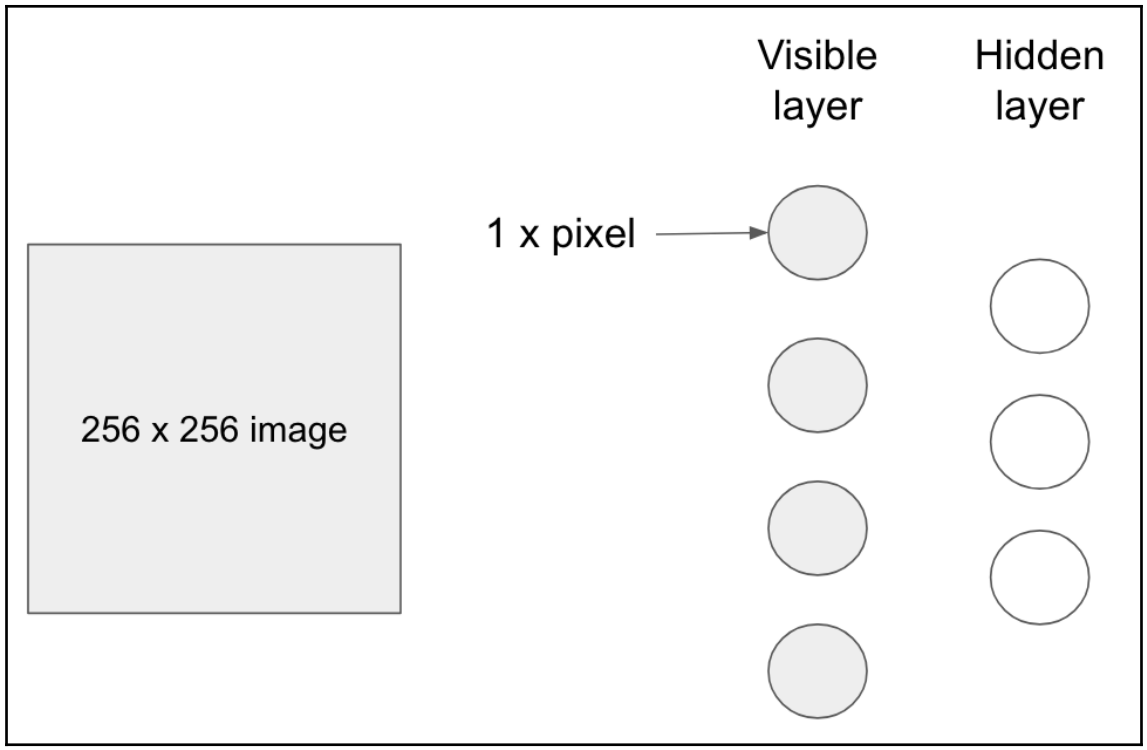

0
50
100
150
200
250
0
50
100
150
200
250

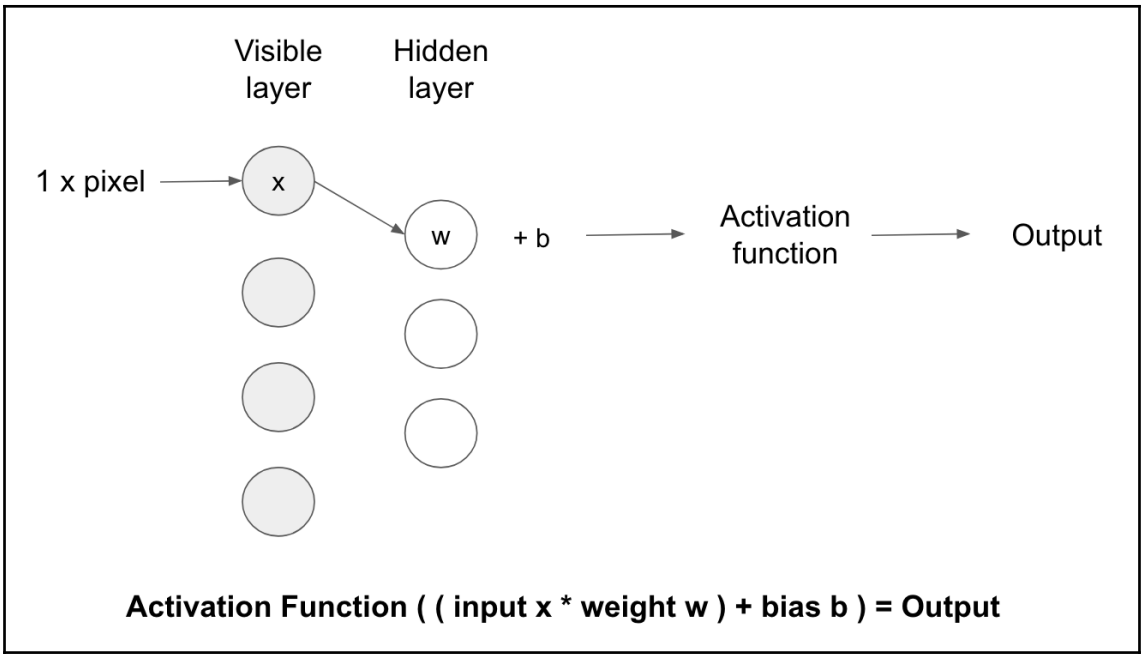
Chapter 9: Deep Belief Networks

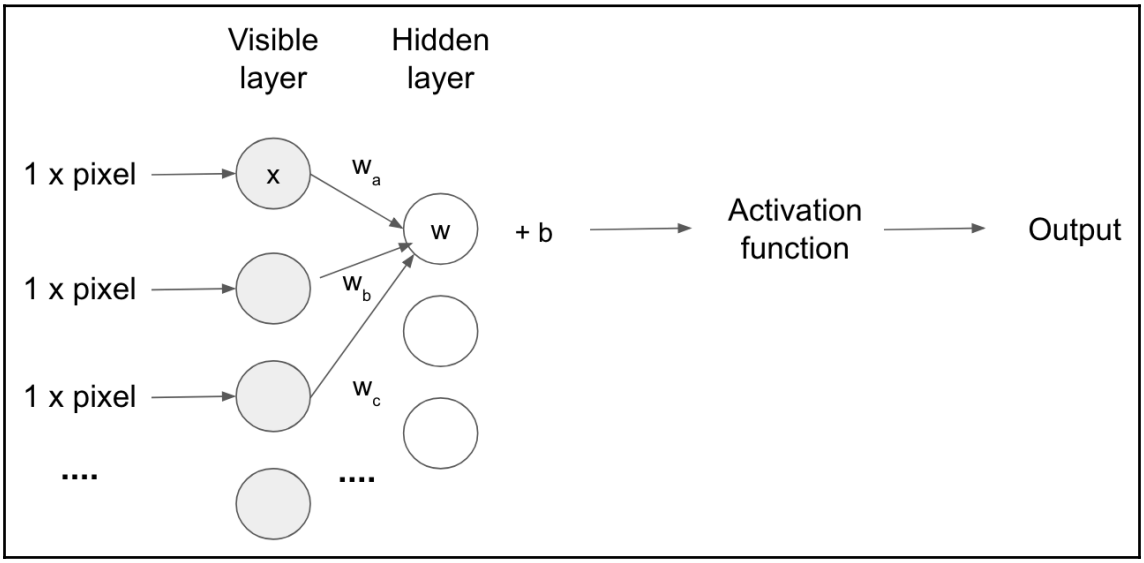


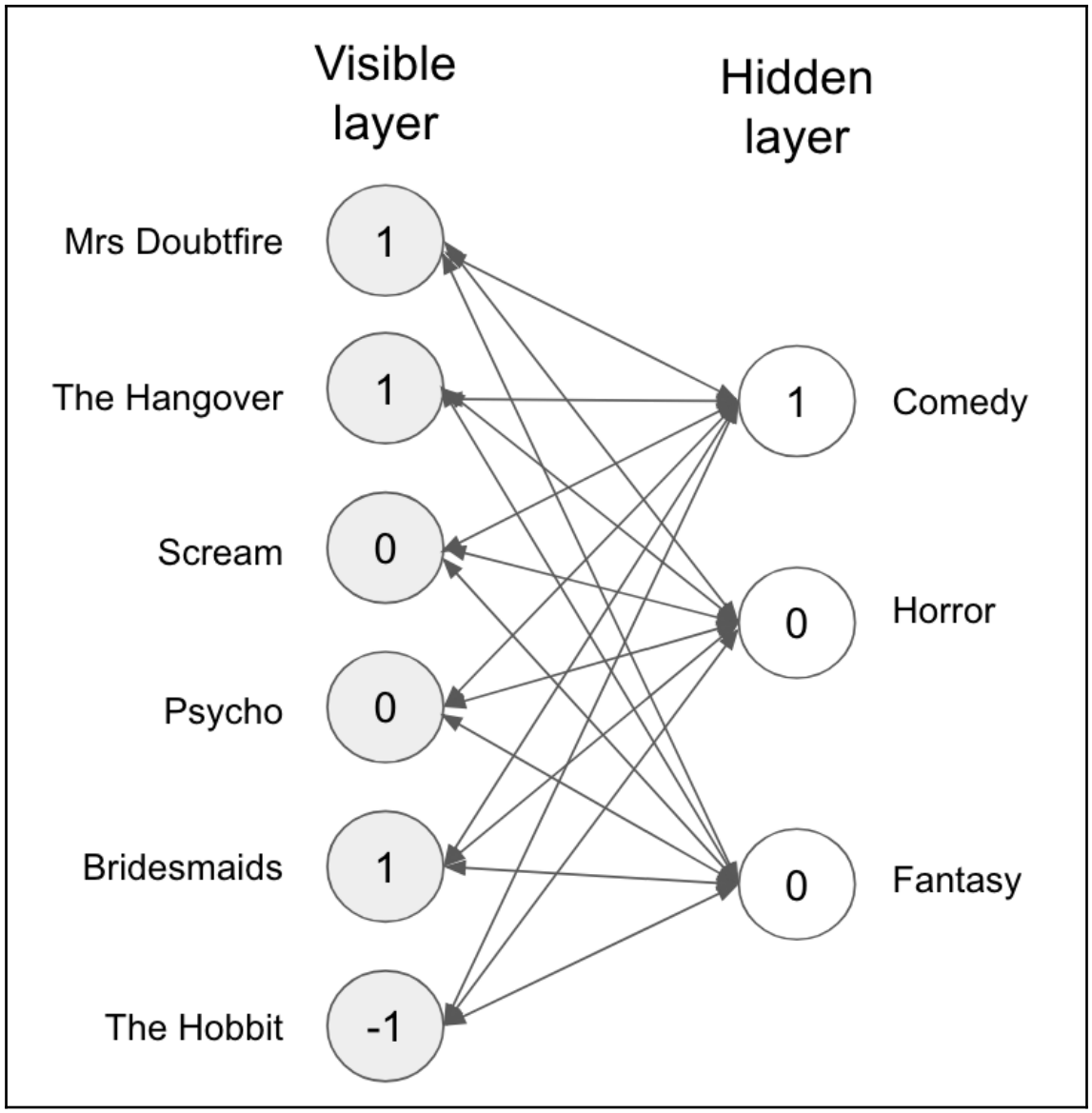

Bayes Network (3 events)
Sunny
Smiling | Sunny
Smiling
Rainbow | Smiling
Rainbow


Visible layer
Hidden layer


Visible layer
Hidden layer
1 x pixel
256 x 256 image

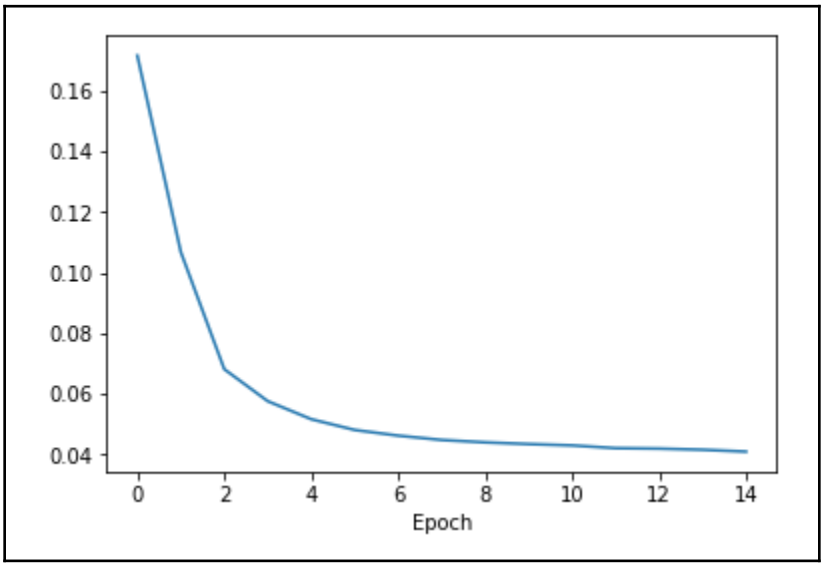

Visible layer
Hidden layer
1 x pixel
x
w
+ b
Activation function
Output
Activation Function ((input x * weight w) + bias b) = Output

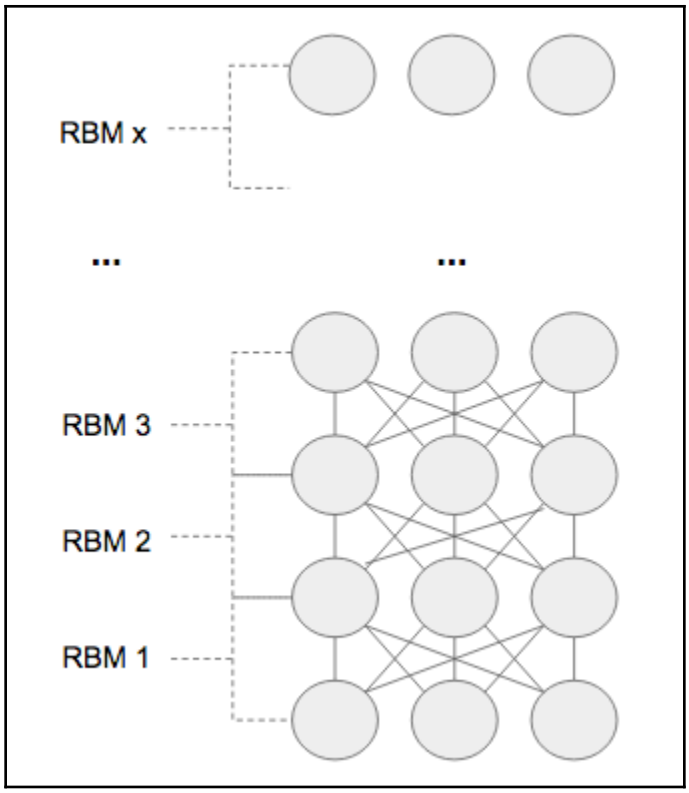

Visible layer
Hidden layer
1 x pixel
1 x pixel
1 x pixel
....
x
w
+ b
Activation function
Output
w_a
w_b
w_c
....

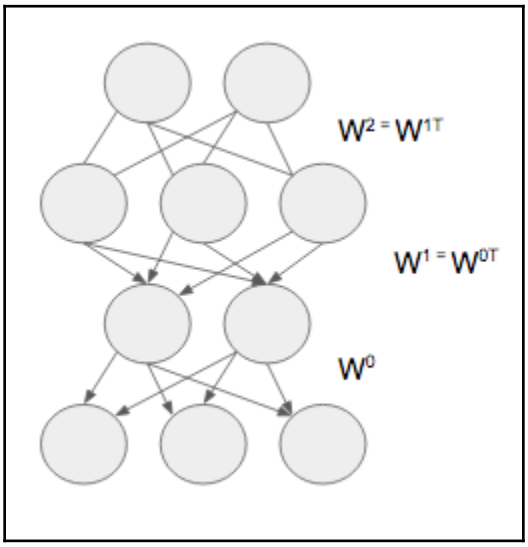


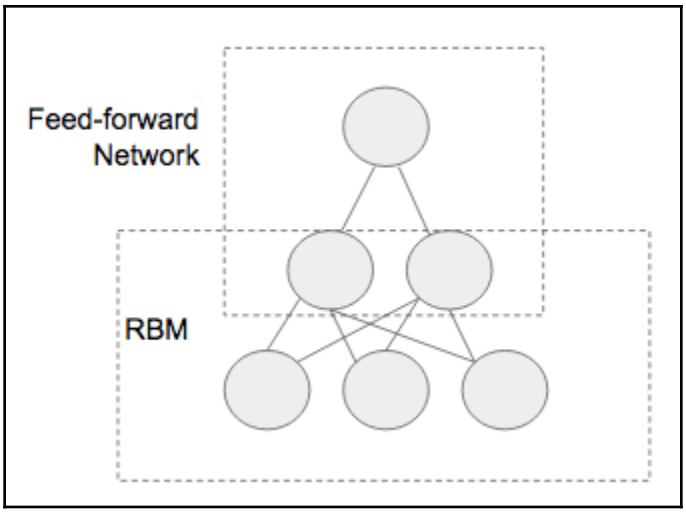
movie_id	List Index	user_id	rating
1	0	1	5
1	0	6	4

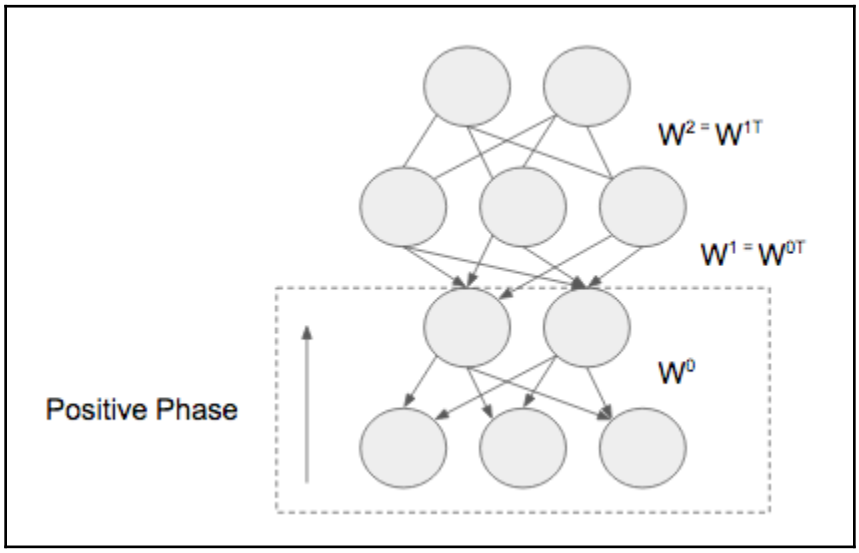
user_id	movie_id	List Index	rating
1	1	0	5
2	21	20	1
3	104	102	4
4	260	257	5
5	6	5	2

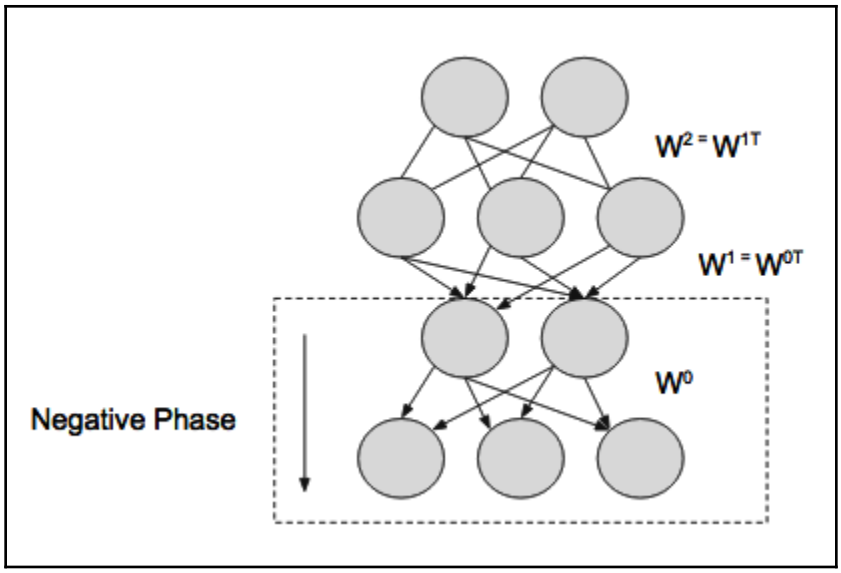


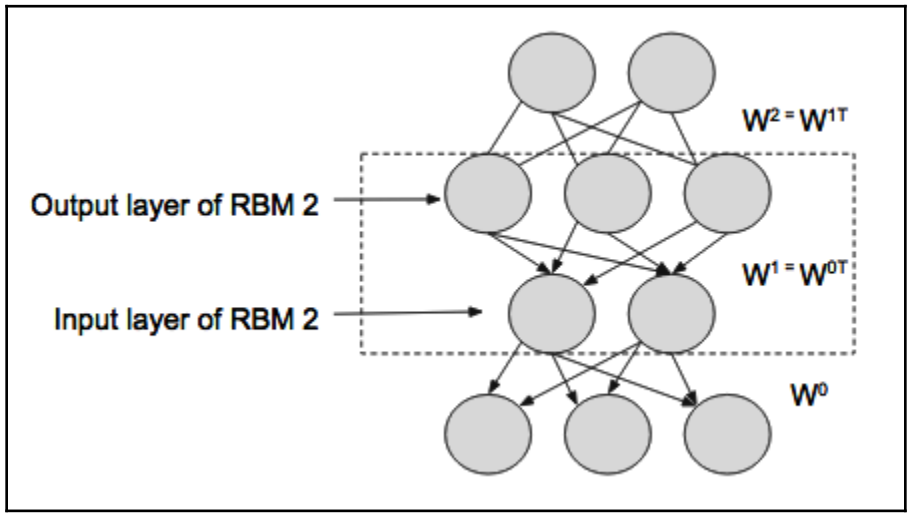

RBM x
...
...
RBM 3
RBM 2
RBM 1

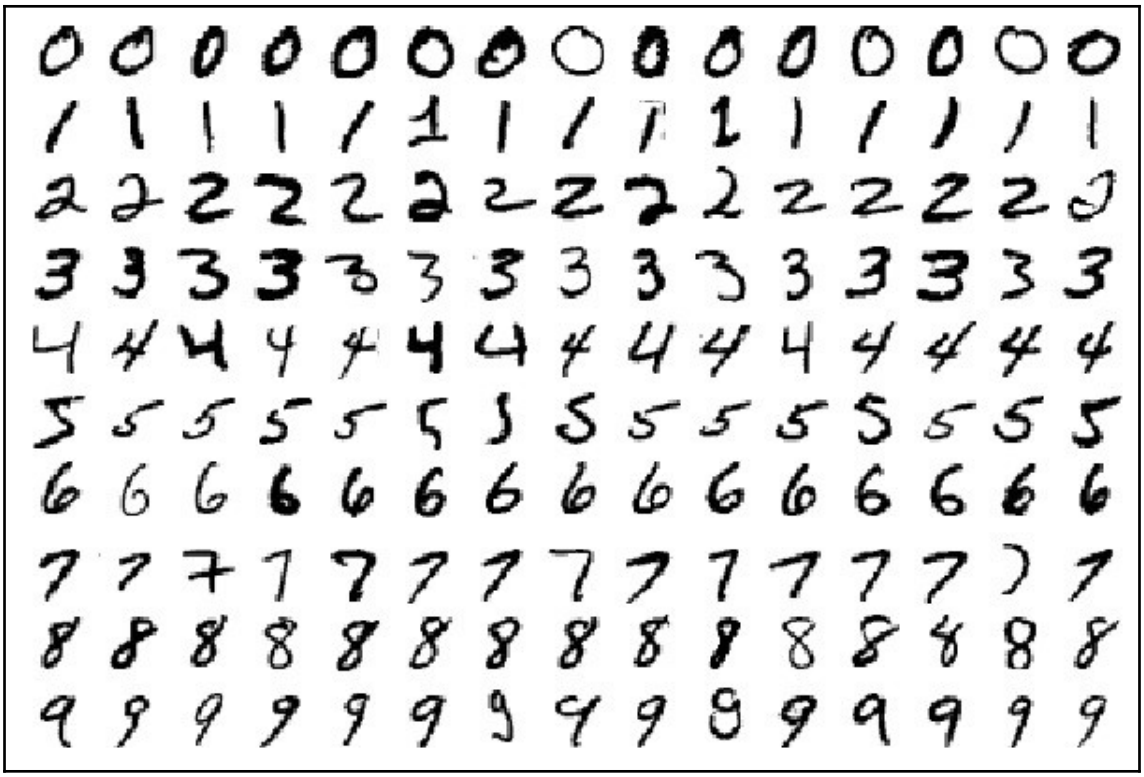

W2 = W1T
W1 = W0T
W0


Feed-forward
Network
RBM


$W^2 = W^{1T}$
$W^1 = W^{0T}$
W^0
Positive Phase

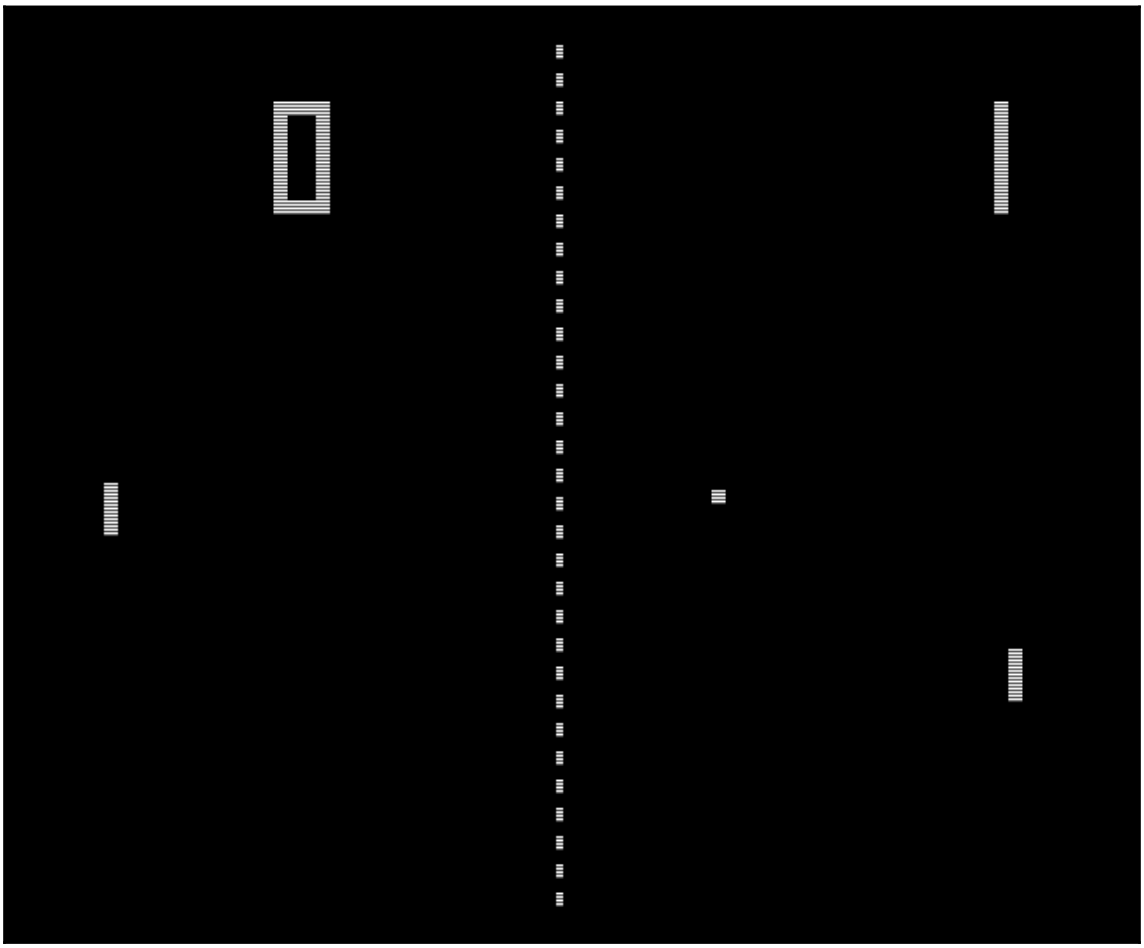

$W^2 = W^{1T}$
$W^1 = W^{0T}$
W^0
Negative Phase

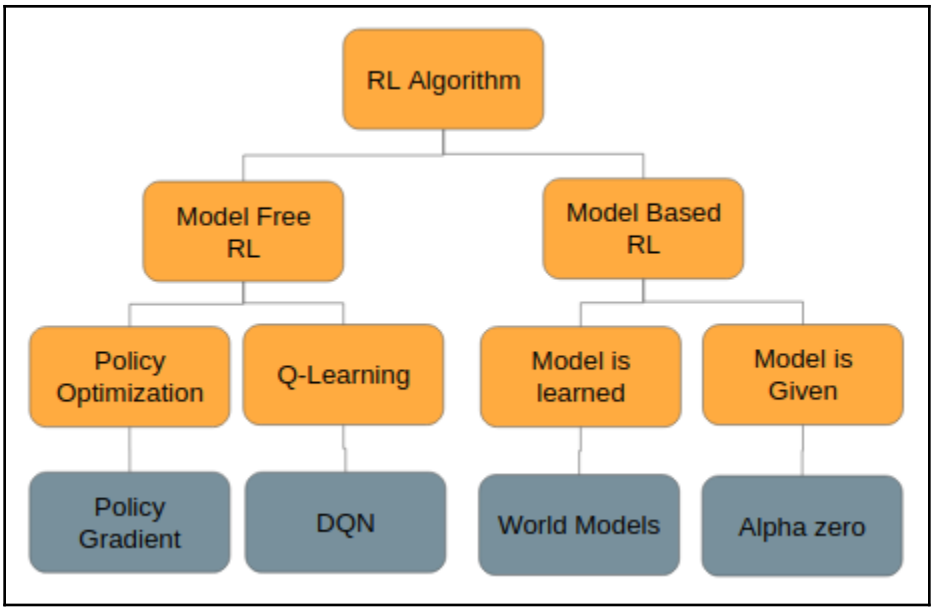

$W^2 = W^{1T}$
Output layer of RBM 2
$W^1 = W^{0T}$
Input layer of RBM 2
W^0

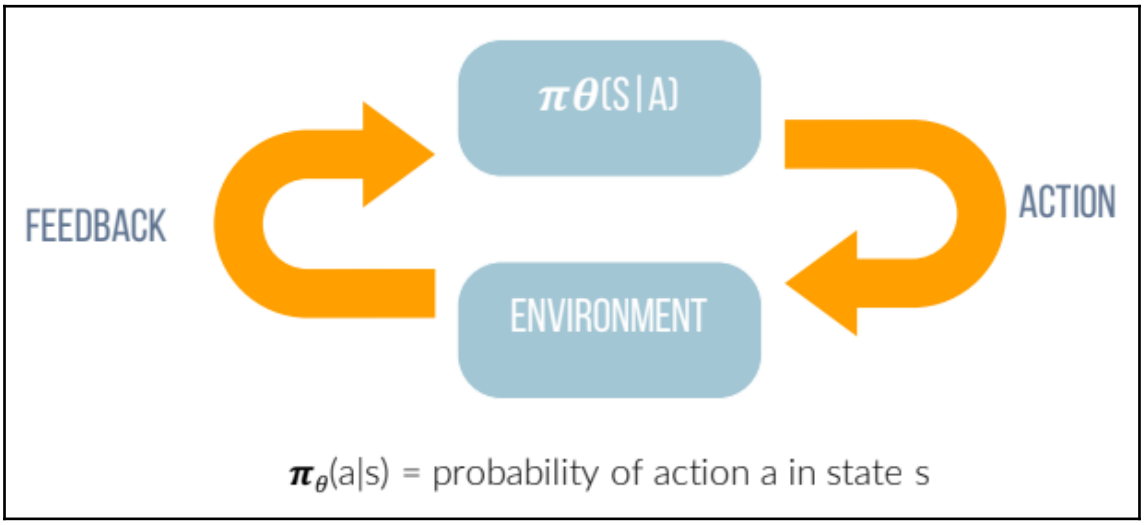


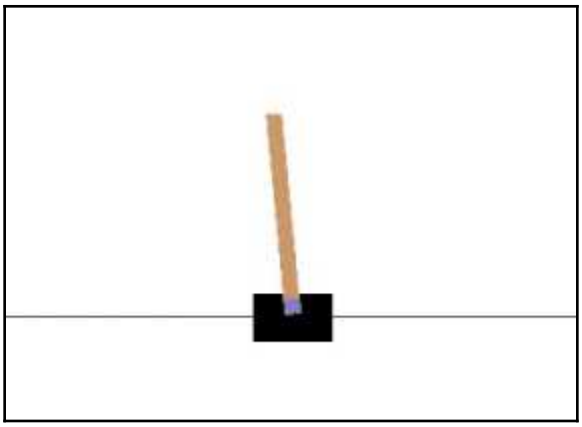
```
Boston house prices dataset
---------------------------

**Data Set Characteristics:**

    :Number of Instances: 506

    :Number of Attributes: 12 numeric/categorical predictive. Median Value (attribute 13) is usually the target.

    :Attribute Information (in order):
        - CRIM     per capita crime rate by town
        - ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
        - INDUS    proportion of non-retail business acres per town
        - CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
        - NOX      nitric oxides concentration (parts per 10 million)
        - RM       average number of rooms per dwelling
        - AGE      proportion of owner-occupied units built prior to 1940
        - DIS      weighted distances to five Boston employment centres
        - RAD      index of accessibility to radial highways
        - TAX      full-value property-tax rate per $10,000
        - PTRATIO  pupil-teacher ratio by town
        - LSTAT    % lower status of the population
        - MEDV     Median value of owner-occupied homes in $1000's

    :Missing Attribute Values: None

    :Creator: Harrison, D. and Rubinfeld, D.L.

This is a copy of UCI ML housing dataset.
https://archive.ics.uci.edu/ml/machine-learning-databases/housing/
```

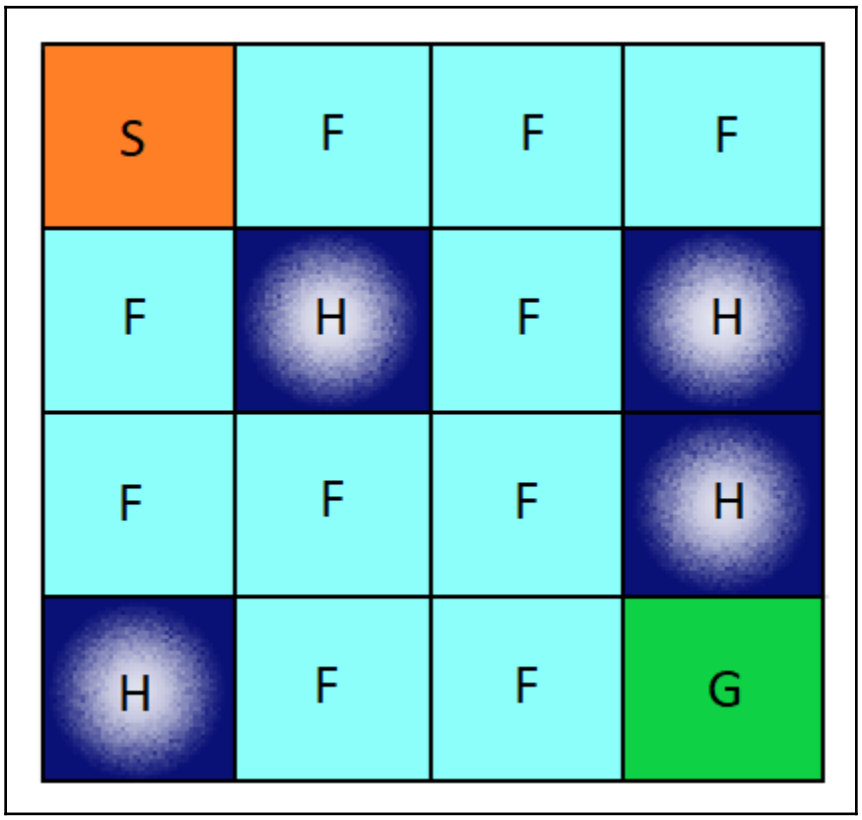
Chapter 10: Reinforcement Learning



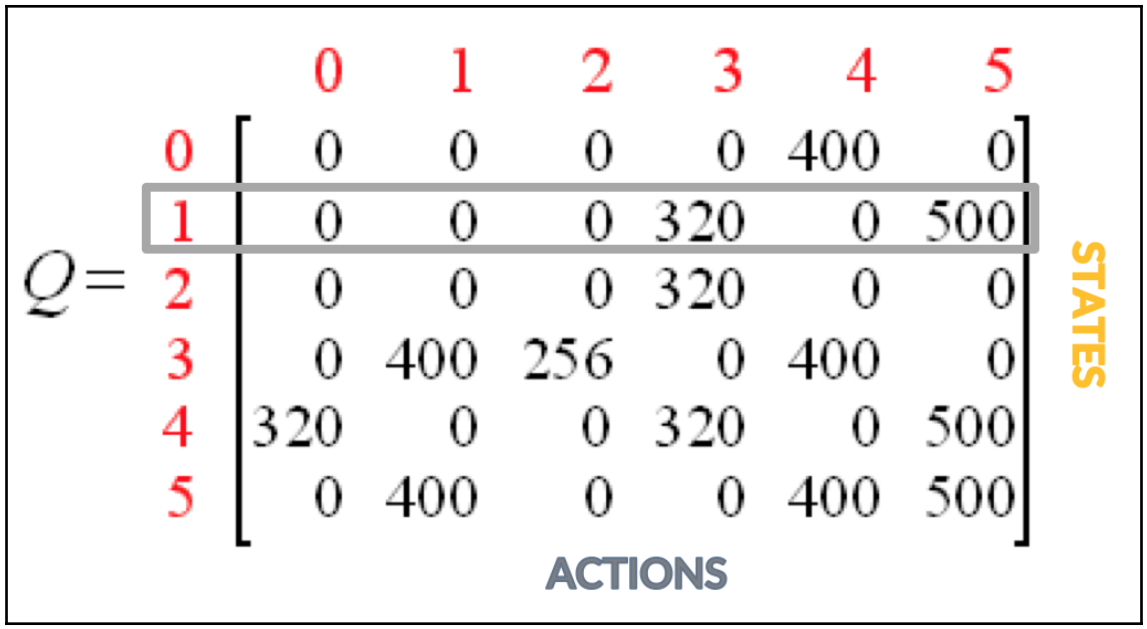

RL Algorithm
Model Free RL
Model Based RL
Policy Optimization
Q-Learning
Model is learned
Model is Given
Policy Gradient
DQN
World Models
Alpha zero

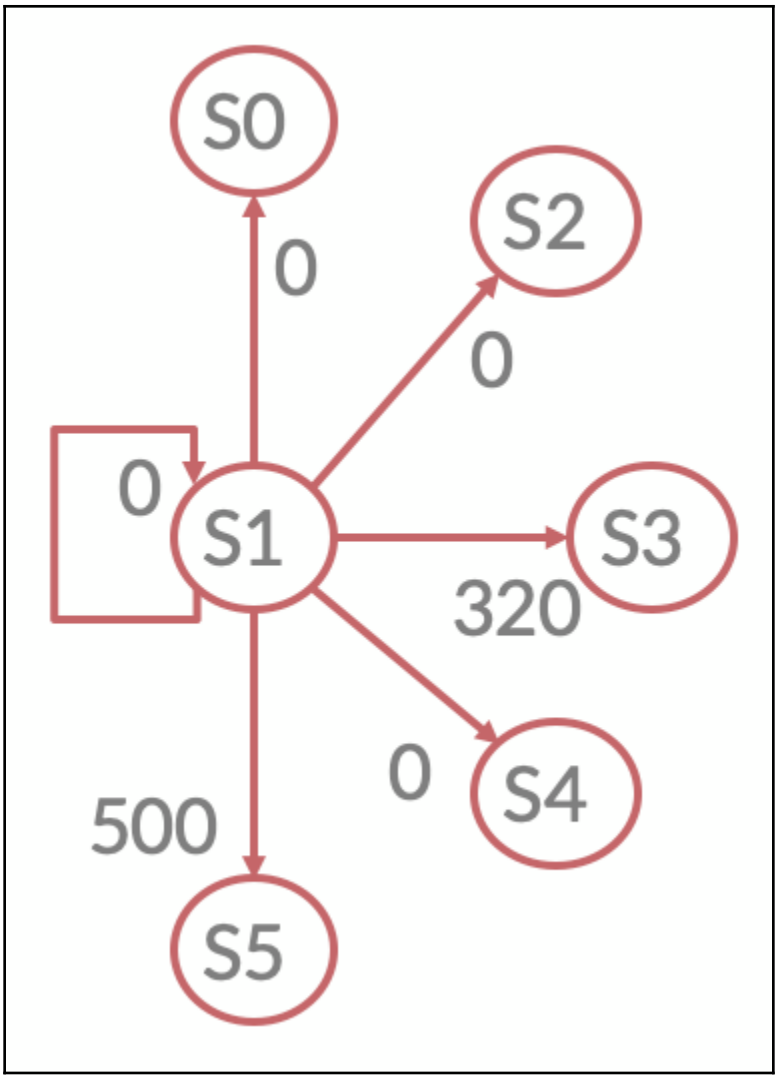

πθ(S|A)
FEEDBACK
ACTION
ENVIRONMENT
πθ(a|s) = probability of action a in state s



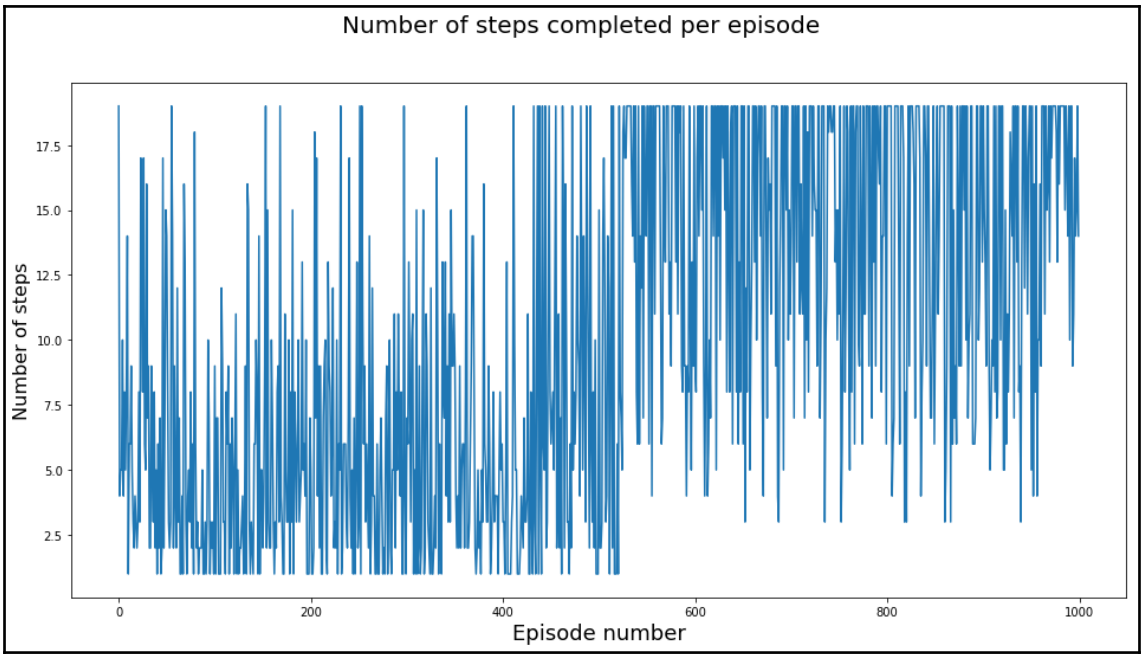

S
F
F
F
F
H
F
H
F
F
F
H
H
F
F
G

```
[[0.00095421 0.00098864 0.00698167 0.00750466]
 [0.00570747 0.00089916 0.00711703 0.00038618]
 [0.00076457 0.00454047 0.00860003 0.00577745]
 [0.00455219 0.00055421 0.00049394 0.00343634]
 [0.00286473 0.00446176 0.00975701 0.00300927]
 [0.00323193 0.00409729 0.0022279  0.00965145]
 [0.00770885 0.00027495 0.00470571 0.00601063]
 [0.00518226 0.00761208 0.00074768 0.00878333]
 [0.00118302 0.00627028 0.00792606 0.0069023 ]
 [0.00330688 0.00721038 0.00506496 0.00677231]
 [0.00541128 0.00174315 0.00387131 0.00637214]
 [0.00548014 0.00976339 0.00628941 0.00262038]
 [0.00733525 0.00279449 0.00077582 0.00691394]
 [0.00079324 0.00387187 0.0059192  0.00177472]
 [0.00299844 0.00402844 0.0062203  0.0023068 ]
 [0.00816794 0.00160594 0.00133737 0.0026781 ]]
```


Q =
0 1 2 3 4 5
0 0 0 0 0 400 0
1 0 0 0 320 0 500
2 0 0 0 320 0 0
3 0 400 256 0 400 0
4 320 0 0 320 0 500
5 0 400 0 0 400 500
STATES
ACTIONS


S0
S2
0
0
0
S1
S3
320
0
S4
500
S5


Reward per episode
Reward
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000
Episode number


Number of steps completed per episode
Number of steps
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0
200
400
600
800
1000
Episode number

Chapter 11: Whats Next?