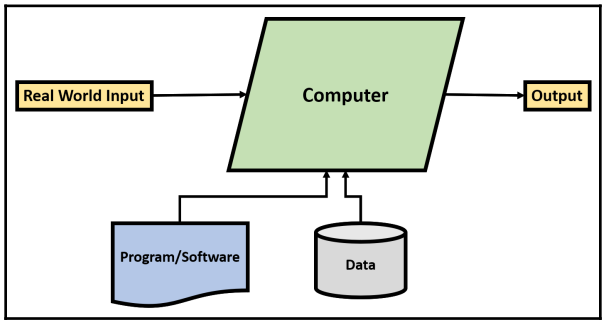
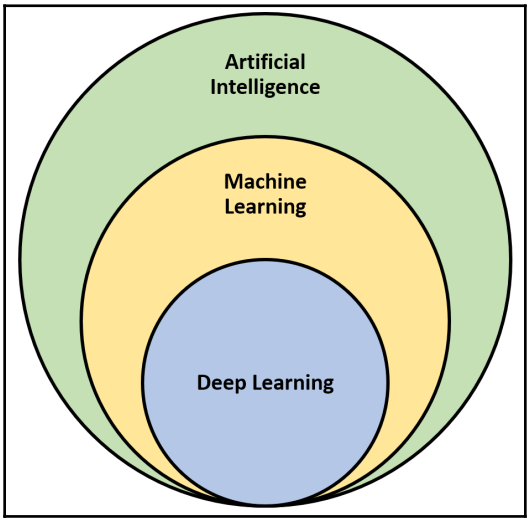
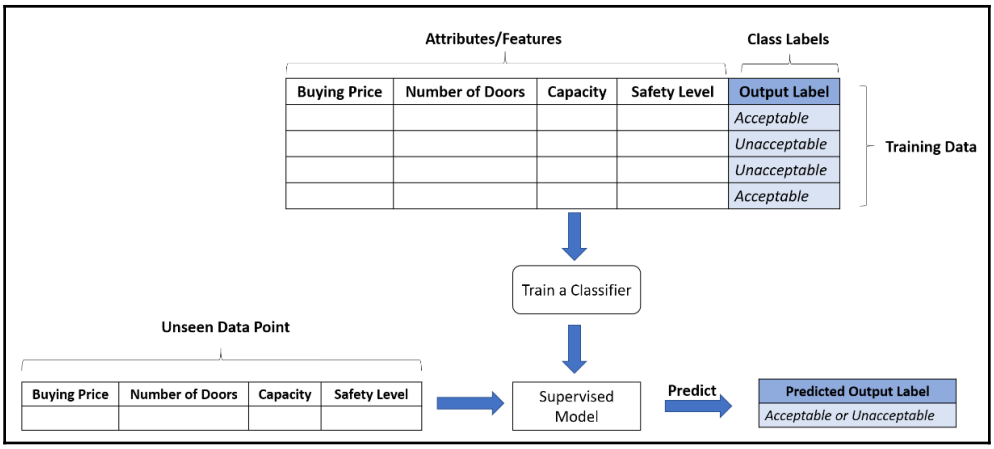
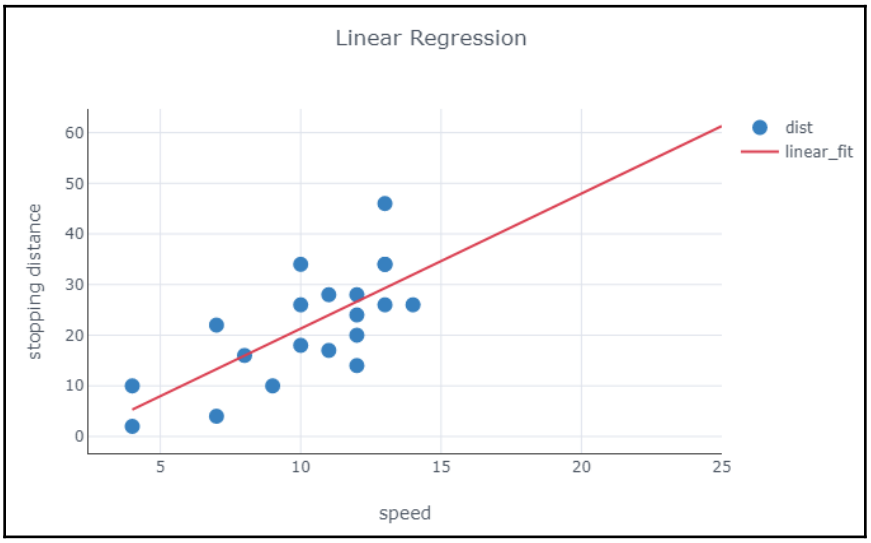
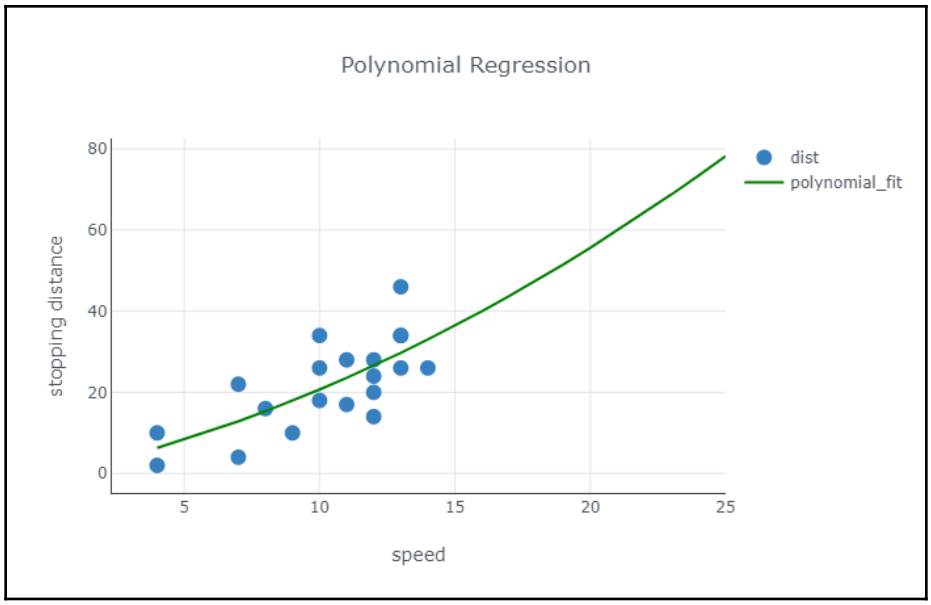
Chapter 1: Machine Learning Fundamentals

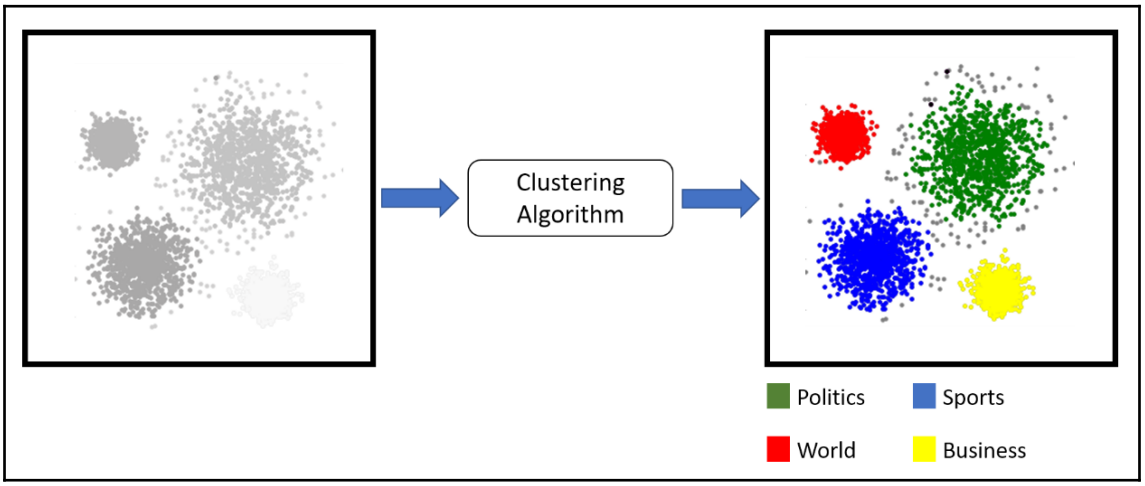


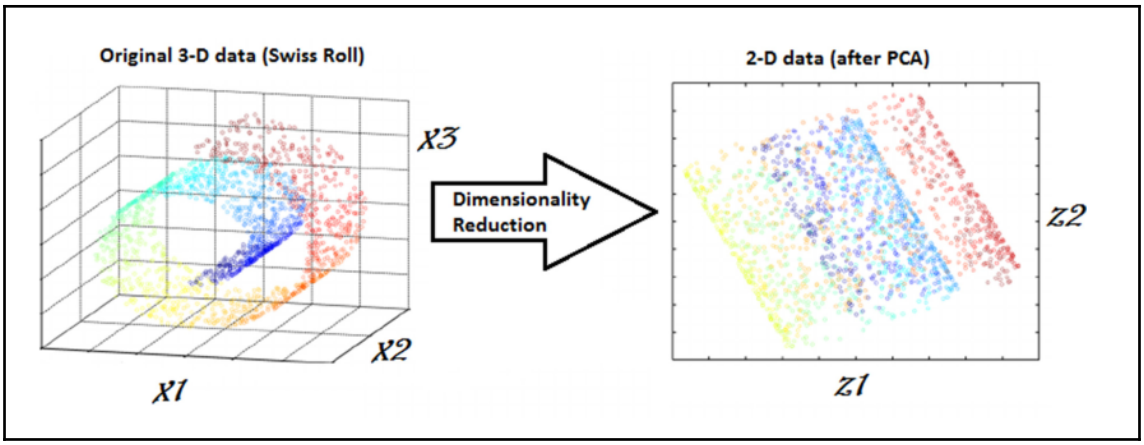


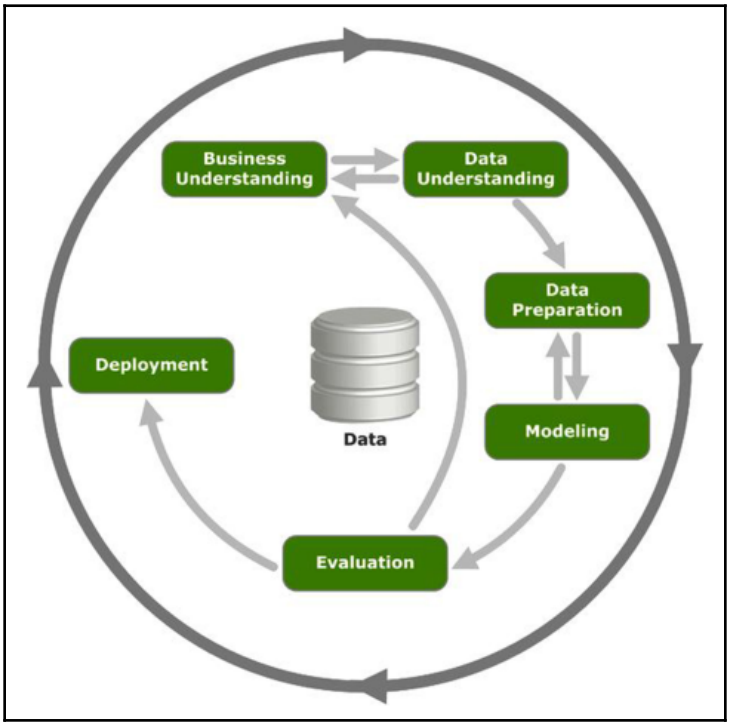

Attributes/Features
Class Labels
Buying Price
Number of Doors
Capacity
Safety Level
Output Label
Acceptable
Unacceptable
Unacceptable
Acceptable
Training Data
Train a Classifier
Unseen Data Point
Buying Price
Number of Doors
Capacity
Safety Level
Supervised Model
Predict
Predicted Output Label
Acceptable or Unacceptable

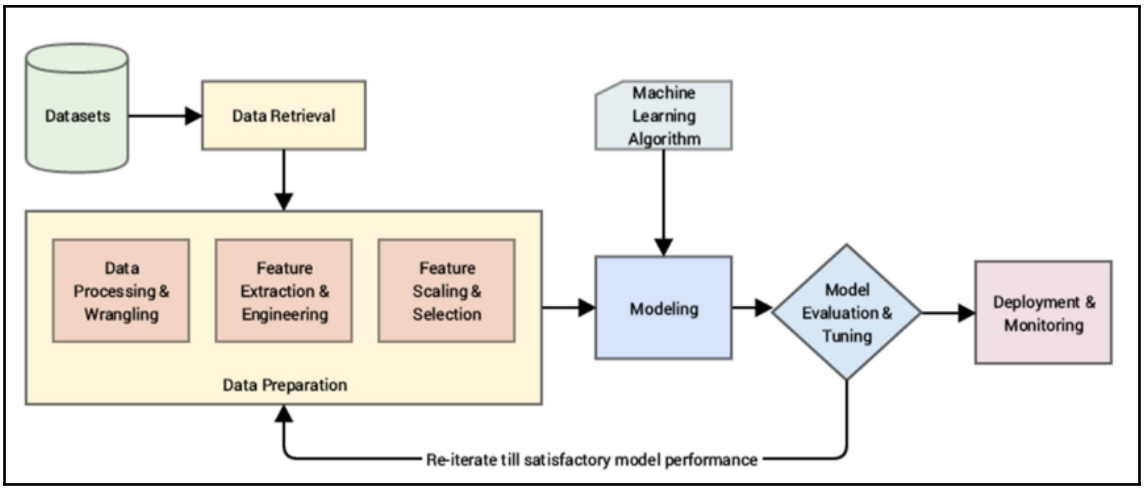

Linear Regression
stopping distance
speed
dist
linear_fit
0
10
20
30
40
50
60
5
10
15
20
25

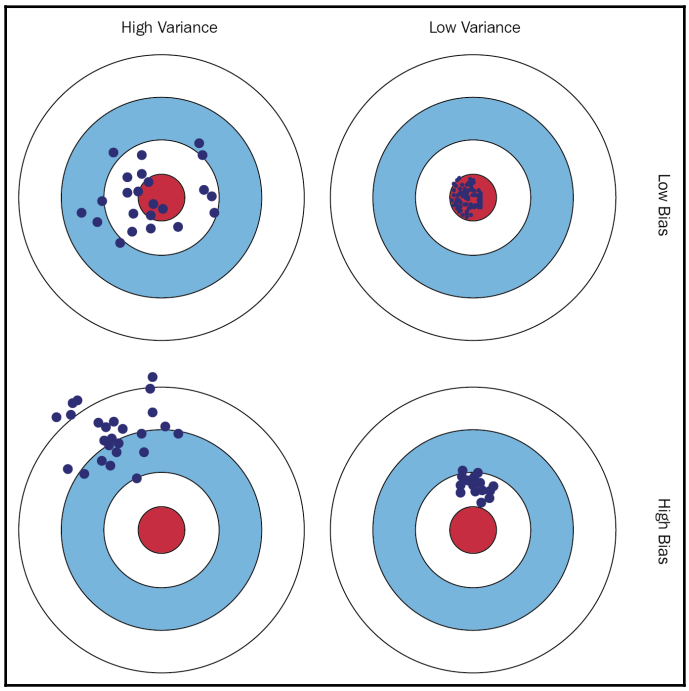

Polynomial Regression
stopping distance
speed
dist
polynomial_fit
0
20
40
60
80
5
10
15
20
25

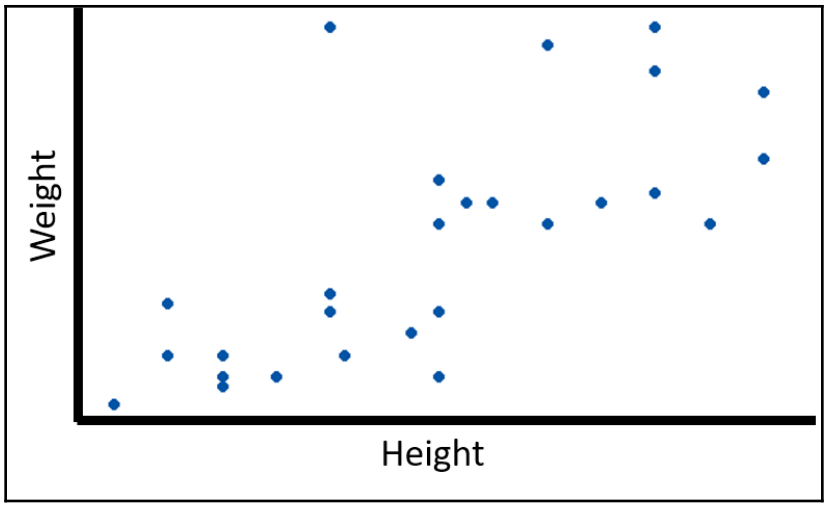

Clustering Algorithm
Politics
Sports
World
Business

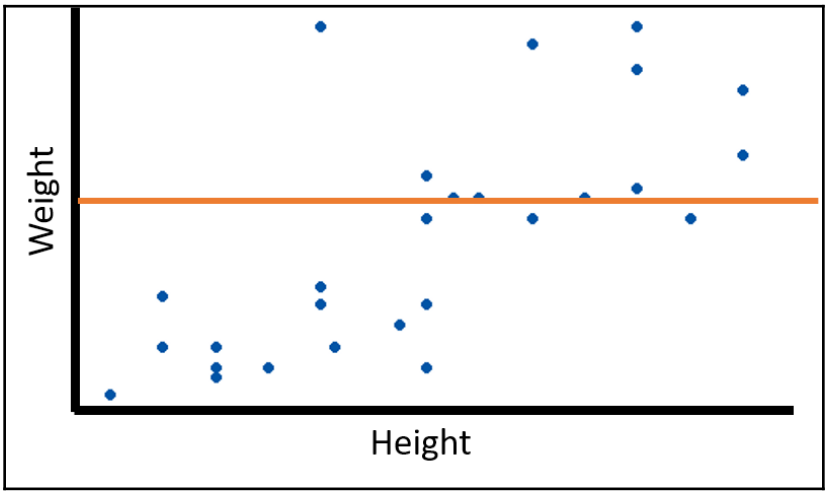

Original 3-D data (Swiss Roll)
X3
X2
X1
Dimensionality Reduction
2-D data (after PCA)
Z2
Z1

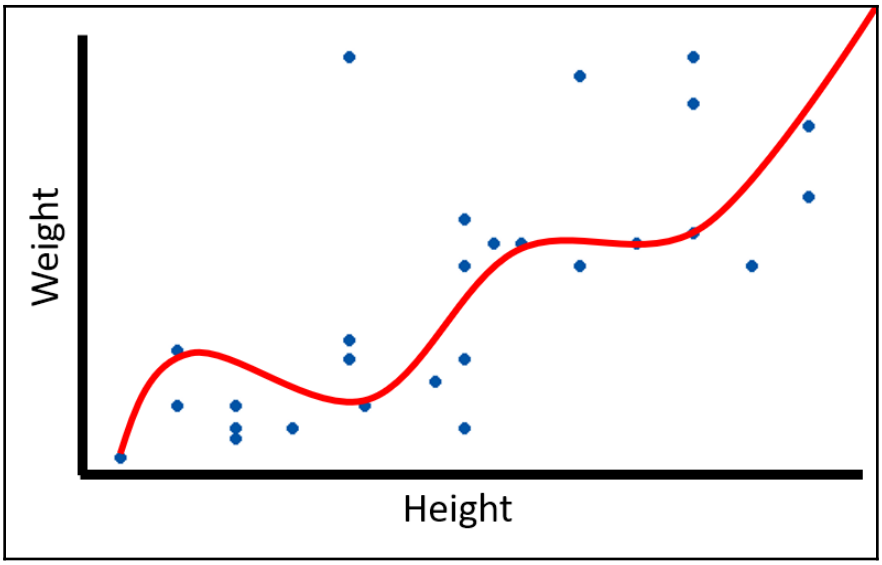

Business Understanding
Data Understanding
Data Preparation
Modeling
Evaluation
Deployment
Data

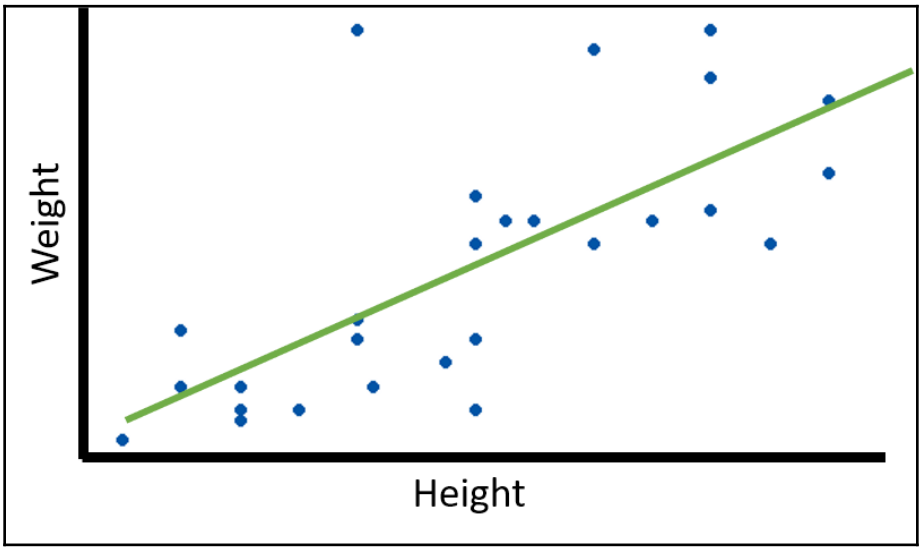

Datasets
Data Retrieval
Machine Learning Algorithm
Data Processing & Wrangling
Feature Extraction & Engineering
Feature Scaling & Selection
Data Preparation
Modeling
Model Evaluation & Tuning
Deployment & Monitoring
Re-iterate till satisfactory model performance


High Variance
Low Variance
Low Bias
High Bias

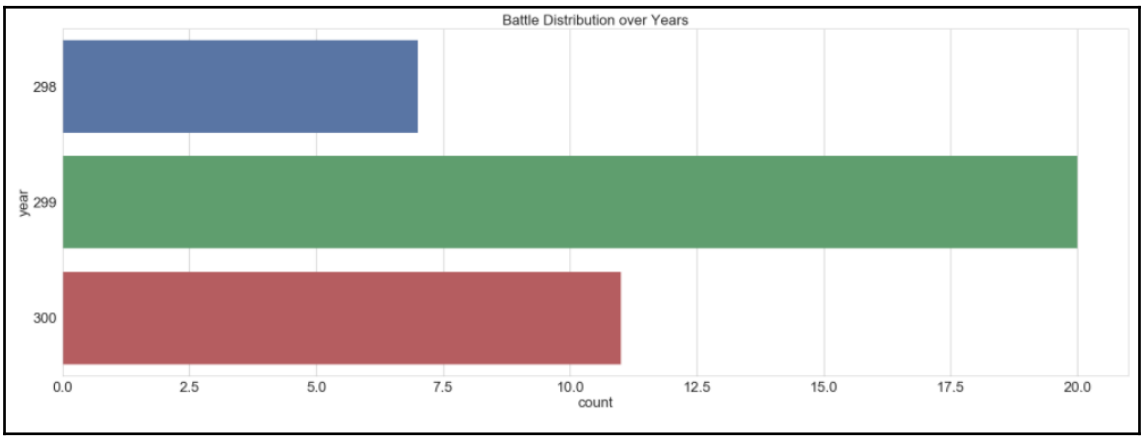

Weight
Height

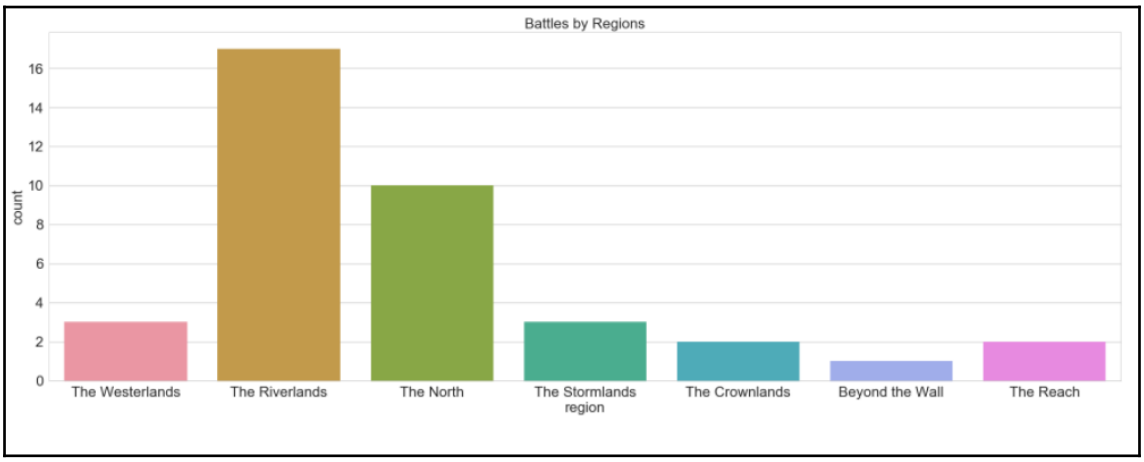

Weight
Height

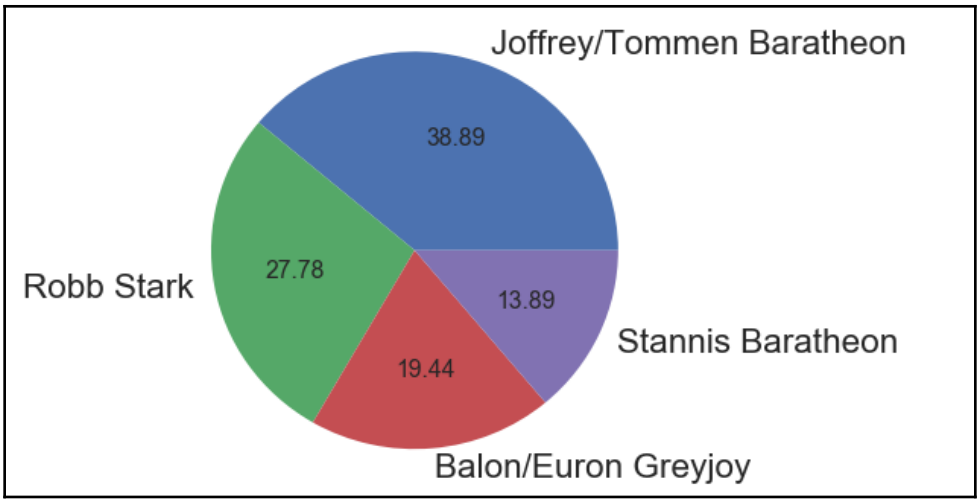

Weight
Height

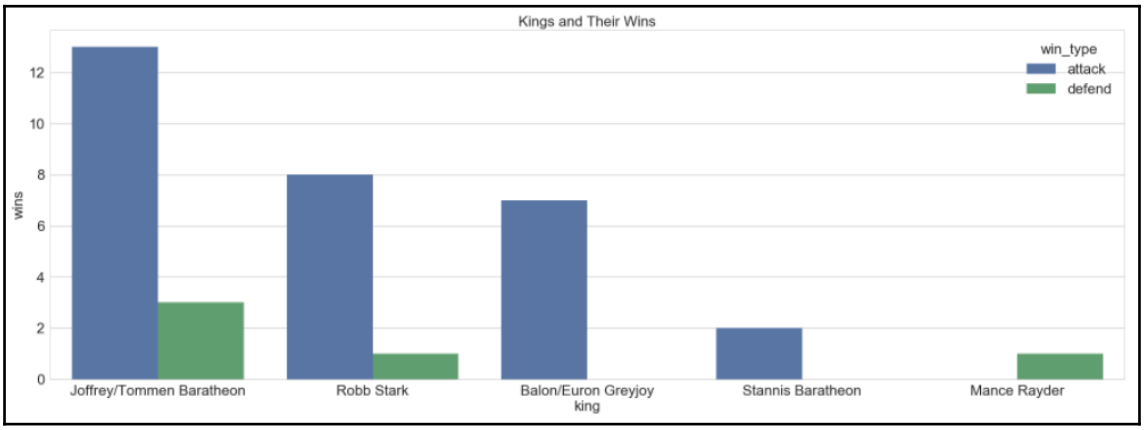

Weight
Height

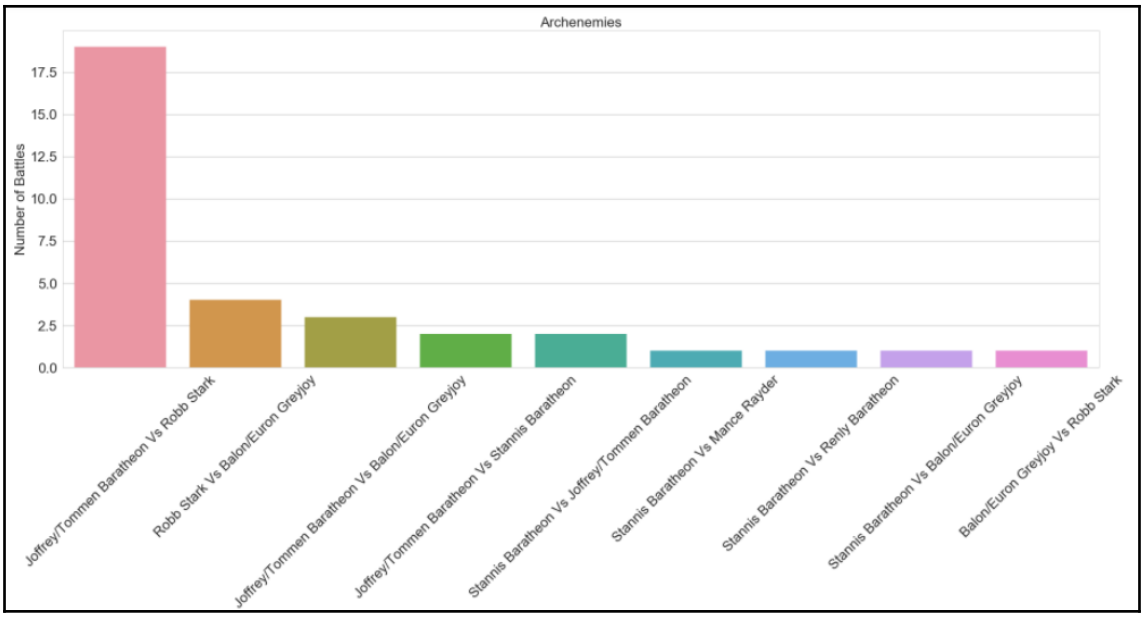
	name	year	battle_number	attacker_king	defender_king	attacker_1	attacker_2	attacker_3	attacker_4	defender_1	...	major_death
0	Battle of the Golden Tooth	298	1	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	NaN	NaN	Tully	...	1.0
1	Battle at the Mummer's Ford	298	2	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	NaN	NaN	Baratheon	...	1.0
2	Battle of Riverrun	298	3	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	NaN	NaN	Tully	...	0.0
3	Battle of the Green Fork	298	4	Robb Stark	Joffrey/Tommen Baratheon	Stark	NaN	NaN	NaN	Lannister	...	1.0
4	Battle of the Whispering Wood	298	5	Robb Stark	Joffrey/Tommen Baratheon	Stark	Tully	NaN	NaN	Lannister	...	1.0



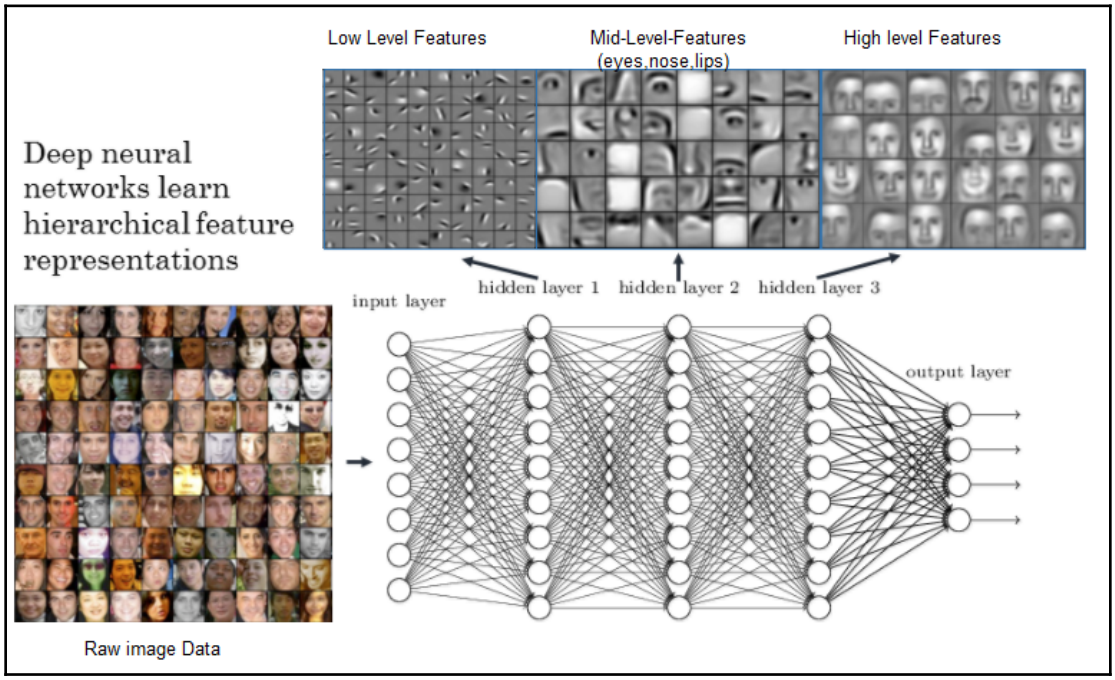


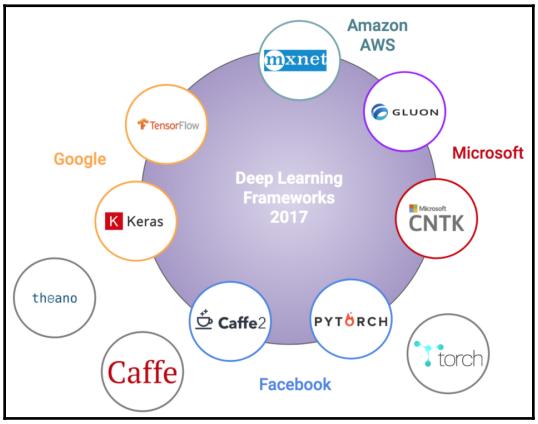

Joffrey/Tommen Baratheon
38.89
Robb Stark
27.78
13.89
Stannis Baratheon
19.44
Balon/Euron Greyjoy

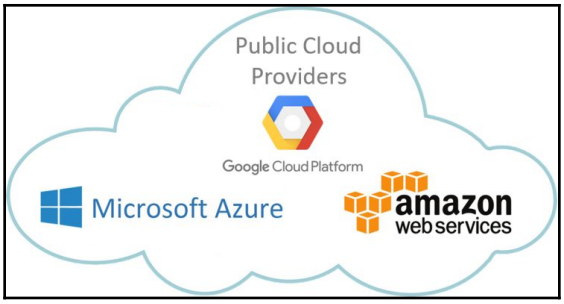

Kings and Their Wins
win_type
attack
defend
wins
0
2
4
6
8
10
12
Joffrey/Tommen Baratheon
Robb Stark
Balon/Euron Greyjoy
king
Stannis Baratheon
Mance Rayder

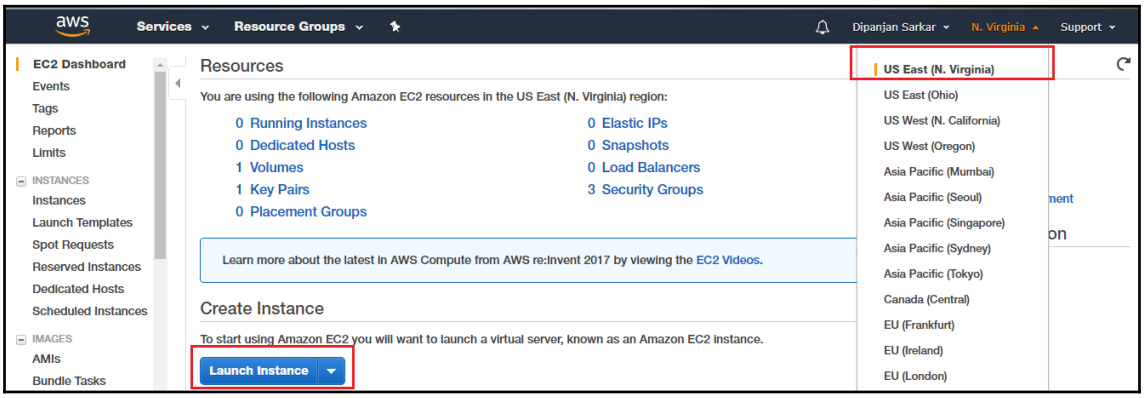

Archenemies
Number of Battles
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Joffrey/Tommen Baratheon Vs Robb Stark
Robb Stark Vs Balon/Euron Greyjoy
Joffrey/Tommen Baratheon Vs Balon/Euron Greyjoy
Joffrey/Tommen Baratheon Vs Stannis Baratheon
Stannis Baratheon Vs Joffrey/Tommen Baratheon
Stannis Baratheon Vs Mance Rayder
Stannis Baratheon Vs Renly Baratheon
Stannis Baratheon Vs Balon/Euron Greyjoy
Balon/Euron Greyjoy Vs Robb Stark

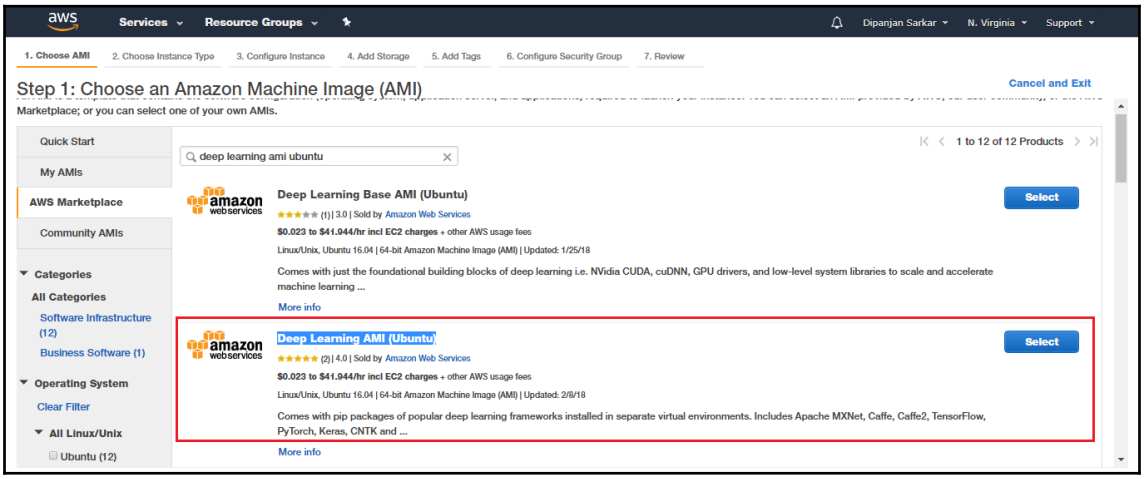
Chapter 2: Deep Learning Essentials

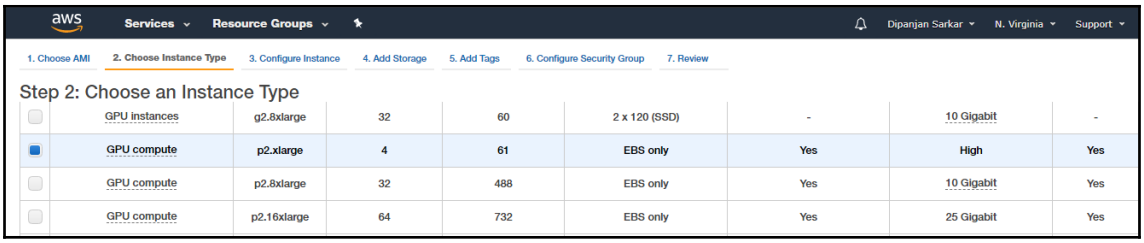


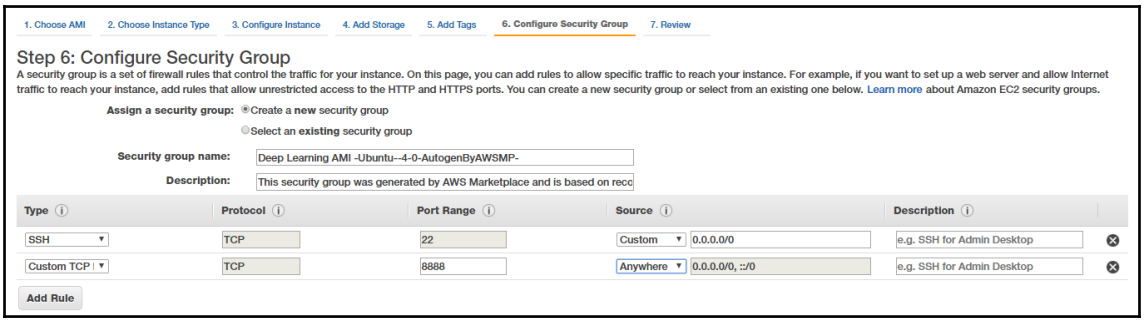


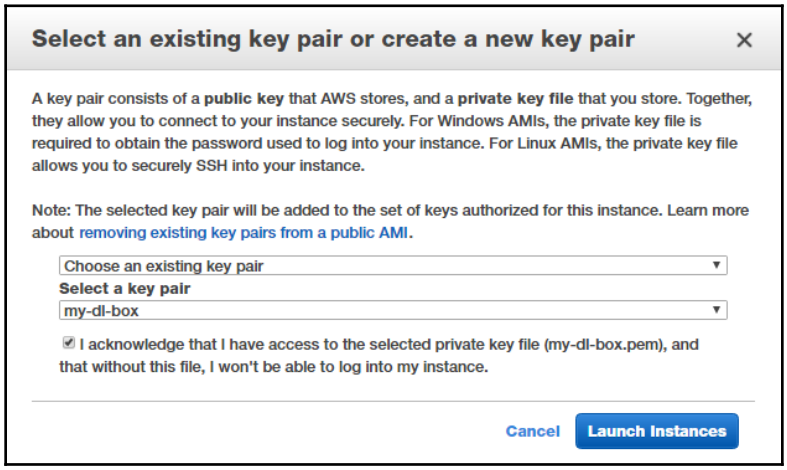

Public Cloud
Providers
Google Cloud Platform
Microsoft Azure
amazon
web services


aws
Services
Resource Groups
Dipanjan Sarkar
N. Virginia
Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Launch Templates
Spot Requests
Reserved Instances
Dedicated Hosts
Scheduled Instances
IMAGES
AMIs
Bundle Tasks
Resources
You are using the following Amazon EC2 resources in the US East (N. Virginia) region:
0 Running Instances
0 Dedicated Hosts
1 Volumes
1 Key Pairs
0 Placement Groups
0 Elastic IPs
0 Snapshots
0 Load Balancers
3 Security Groups
Learn more about the latest in AWS Compute from AWS re:Invent 2017 by viewing the EC2 Videos.
Create Instance
To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.
Launch Instance
US East (N. Virginia)
US East (Ohio)
US West (N. California)
US West (Oregon)
Asia Pacific (Mumbai)
Asia Pacific (Seoul)
Asia Pacific (Singapore)
Asia Pacific (Sydney)
Asia Pacific (Tokyo)
Canada (Central)
EU (Frankfurt)
EU (Ireland)
EU (London)

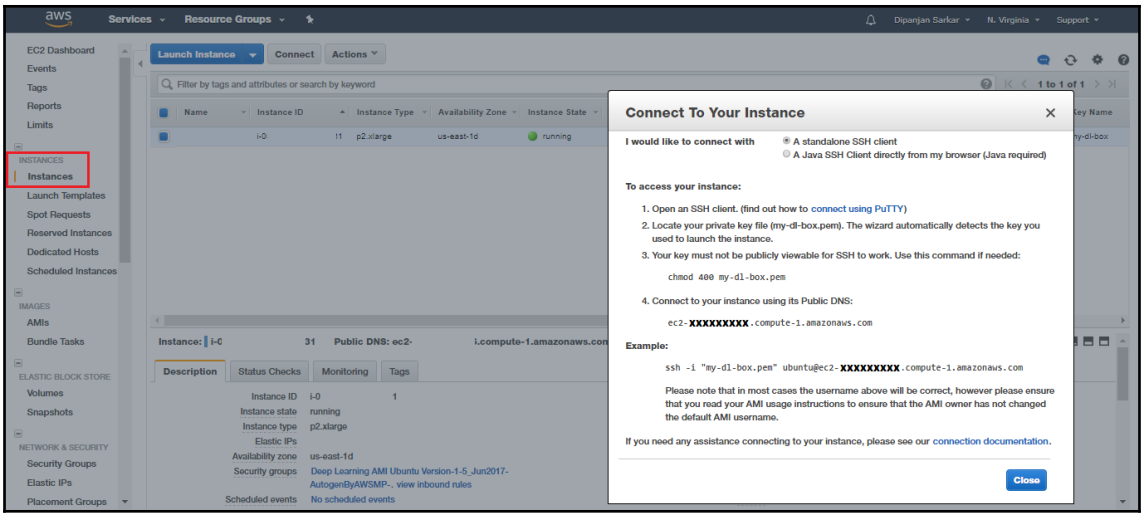

aws
Services
Resource Groups
Dipanjan Sarkar
N. Virginia
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 1: Choose an Amazon Machine Image (AMI)
Cancel and Exit
Marketplace; or you can select one of your own AMIs.
Quick Start
My AMIs
AWS Marketplace
Community AMIs
Categories
All Categories
Software Infrastructure (12)
Business Software (1)
Operating System
Clear Filter
All Linux/Unix
Ubuntu (12)
deep learning ami ubuntu
1 to 12 of 12 Products
Deep Learning Base AMI (Ubuntu)
Sold by Amazon Web Services
$0.023 to $41.944/hr incl EC2 charges + other AWS usage fees
Linux/Unix, Ubuntu 16.04 | 64-bit Amazon Machine Image (AMI) | Updated: 1/25/18
Comes with just the foundational building blocks of deep learning i.e. NVidia CUDA, cuDNN, GPU drivers, and low-level system libraries to scale and accelerate machine learning ...
More info
Select
Deep Learning AMI (Ubuntu)
Sold by Amazon Web Services
$0.023 to $41.944/hr incl EC2 charges + other AWS usage fees
Linux/Unix, Ubuntu 16.04 | 64-bit Amazon Machine Image (AMI) | Updated: 2/8/18
Comes with pip packages of popular deep learning frameworks installed in separate virtual environments. Includes Apache MXNet, Caffe, Caffe2, TensorFlow, PyTorch, Keras, CNTK and ...
More info
Select


aws
Services
Resource Groups
Dipanjan Sarkar
N. Virginia
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 2: Choose an Instance Type
GPU instances g2.8xlarge 32 60 2 x 120 (SSD) - 10 Gigabit -
GPU compute p2.xlarge 4 61 EBS only Yes High Yes
GPU compute p2.8xlarge 32 488 EBS only Yes 10 Gigabit Yes
GPU compute p2.16xlarge 64 732 EBS only Yes 25 Gigabit Yes

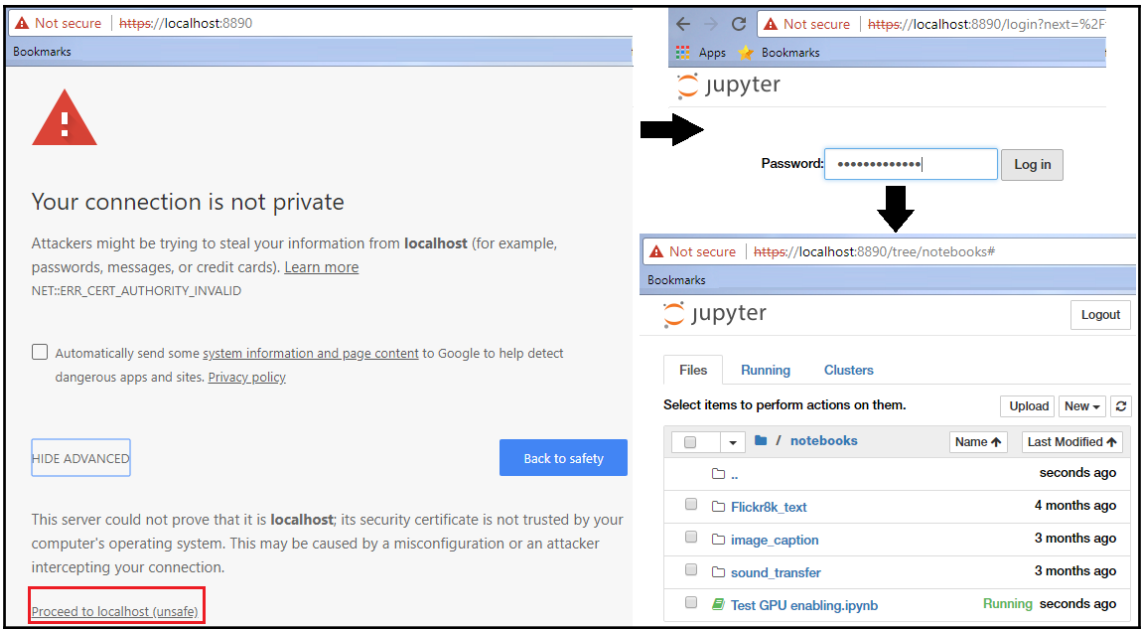

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 6: Configure Security Group
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.
Assign a security group: Create a new security group
Select an existing security group
Security group name: Deep Learning AMI -Ubuntu--4-0-AutogenByAWSMP-
Description: This security group was generated by AWS Marketplace and is based on recc
Type Protocol Port Range Source Description
SSH TCP 22 Custom 0.0.0.0/0 e.g. SSH for Admin Desktop
Custom TCP TCP 8888 Anywhere 0.0.0.0/0, ::/0 e.g. SSH for Admin Desktop
Add Rule

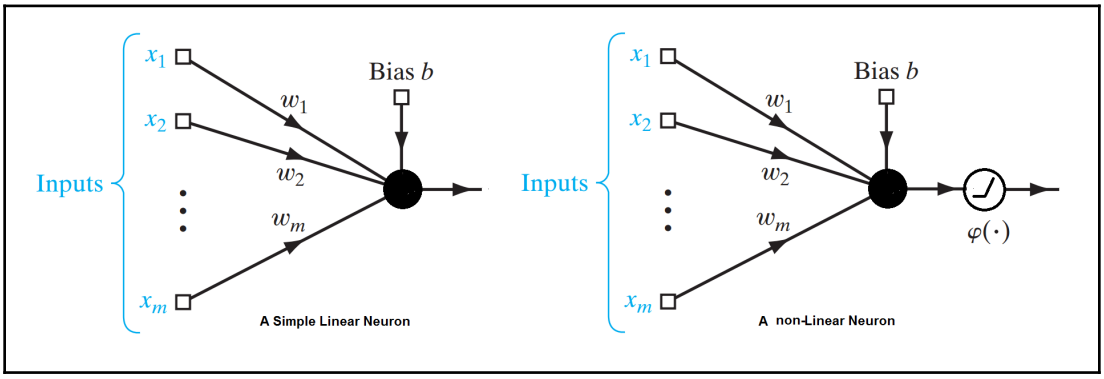

Select an existing key pair or create a new key pair
A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.
Choose an existing key pair
Select a key pair
my-dl-box
I acknowledge that I have access to the selected private key file (my-dl-box.pem), and that without this file, I won't be able to log into my instance.
Cancel
Launch Instances

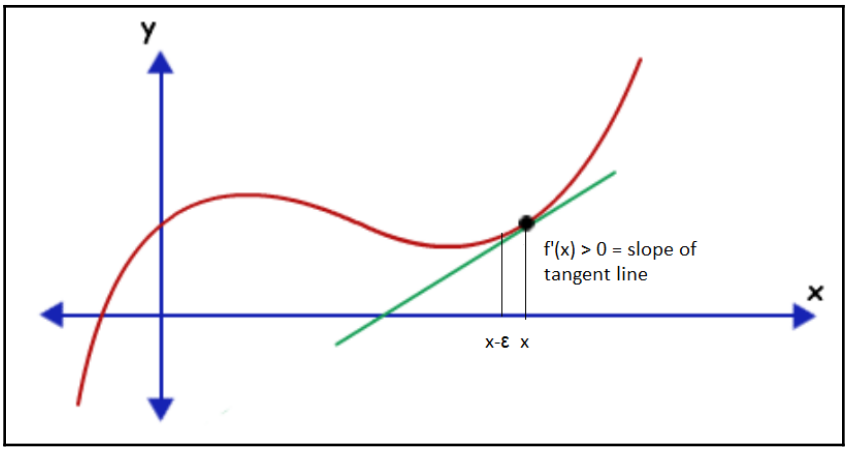

aws
Services
Resource Groups
Dipanjan Sarkar
N. Virginia
Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Running On-Demand p2.8xlarge instances 1 Request limit increase
Running On-Demand p2.xlarge instances 1 Request limit increase
Running On-Demand p3.16xlarge instances 0 Request limit increase
Running On-Demand p3.2xlarge instances 1 Request limit increase
Running On-Demand p3.8xlarge instances 1 Request limit increase

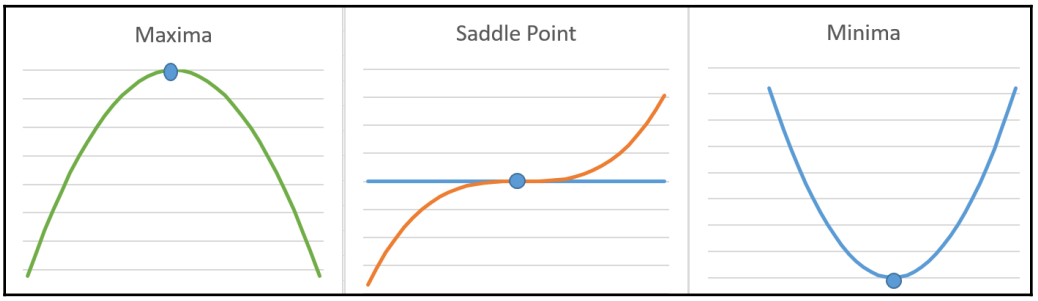


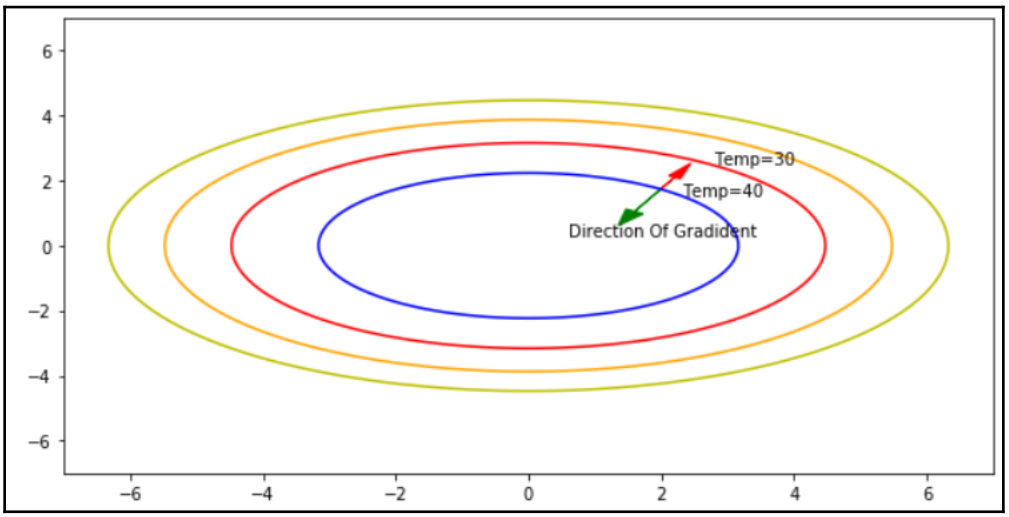

```
Sun Feb 25 15:18:59 2018
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 384.66                 Driver Version: 384.66                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla K80           On   | 00000000:00:1E.0 Off |                    0 |
| N/A   40C    P8    31W / 149W |      0MiB / 11439MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

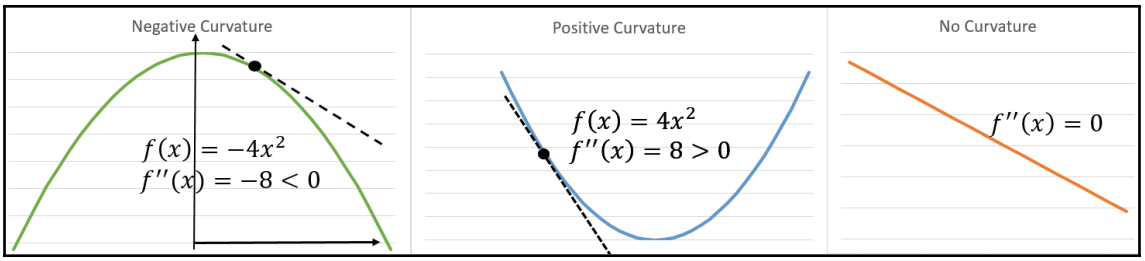

Not secure | https://localhost:8890
Bookmarks
Your connection is not private
Attackers might be trying to steal your information from localhost (for example, passwords, messages, or credit cards). Learn more
NET::ERR_CERT_AUTHORITY_INVALID
Automatically send some system information and page content to Google to help detect dangerous apps and sites. Privacy policy
HIDE ADVANCED
Back to safety
This server could not prove that it is localhost; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an attacker intercepting your connection.
Proceed to localhost (unsafe)
Not secure | https://localhost:8890/login?next=%2F
Apps
Bookmarks
jupyter
Password:
Log in
Not secure | https://localhost:8890/tree/notebooks#
Bookmarks
jupyter
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
/ notebooks
Name
Last Modified
seconds ago
Flickr8k_text
4 months ago
image_caption
3 months ago
sound_transfer
3 months ago
Test GPU enabling.ipynb
Running seconds ago

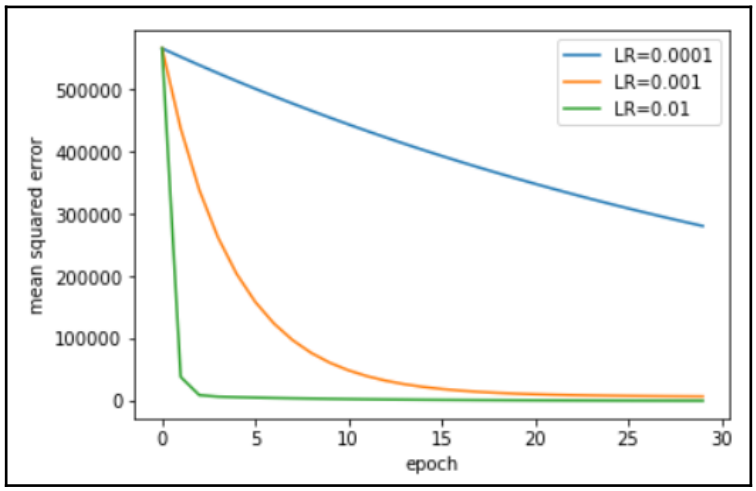

Inputs
x_1
x_2
x_m
w_1
w_2
w_m
Bias b
A Simple Linear Neuron
Inputs
x_1
x_2
x_m
w_1
w_2
w_m
Bias b
$\varphi(\cdot)$
A non-Linear Neuron

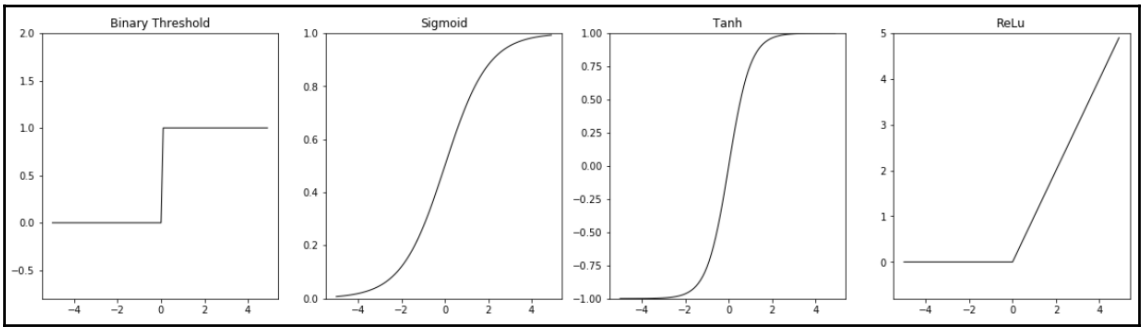

y
x
f'(x) > 0 = slope of
tangent line
x-ε x

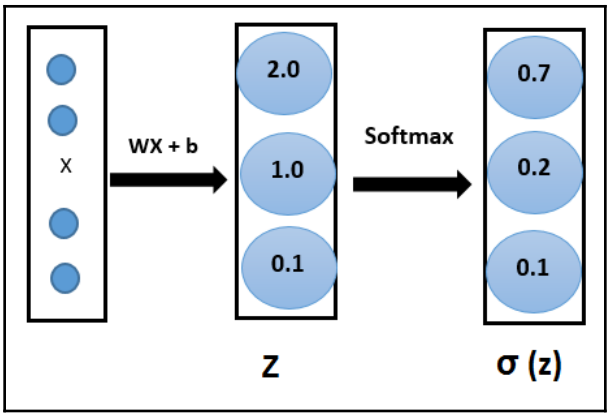

Maxima
Saddle Point
Minima

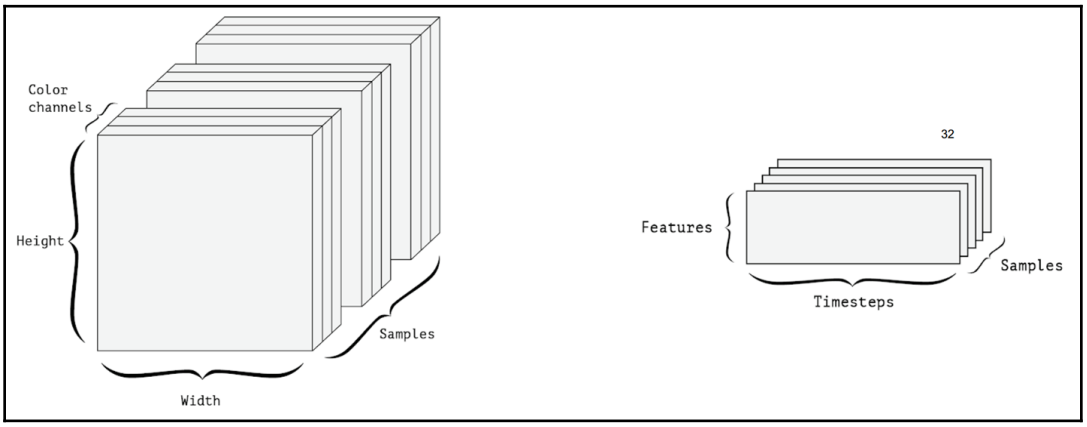

Temp=30
Temp=40
Direction Of Gradident
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6

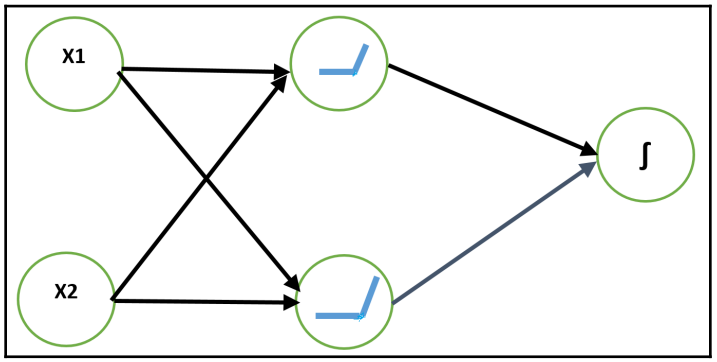

Negative Curvature
$f(x) = -4x^2$
$f''(x) = -8 < 0$
Positive Curvature
$f(x) = 4x^2$
$f''(x) = 8 > 0$
No Curvature
$f''(x) = 0$

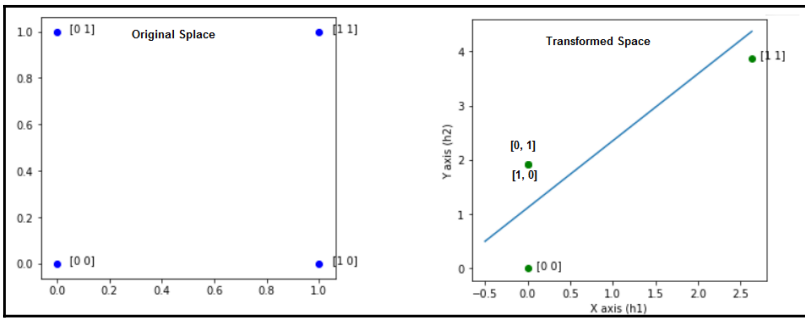

LR=0.0001
LR=0.001
LR=0.01
mean squared error
epoch
500000
400000
300000
200000
100000
0
0
5
10
15
20
25
30


Binary Threshold
Sigmoid
Tanh
ReLu


X
WX + b
2.0
1.0
0.1
Softmax
0.7
0.2
0.1
Z
σ (z)

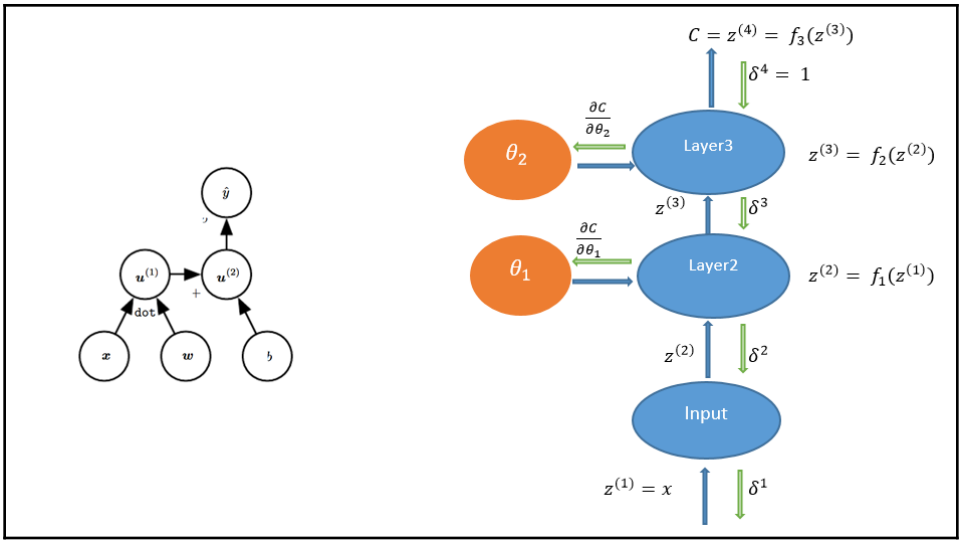

Color
channels
Height
Width
Samples
32
Features
Timesteps
Samples

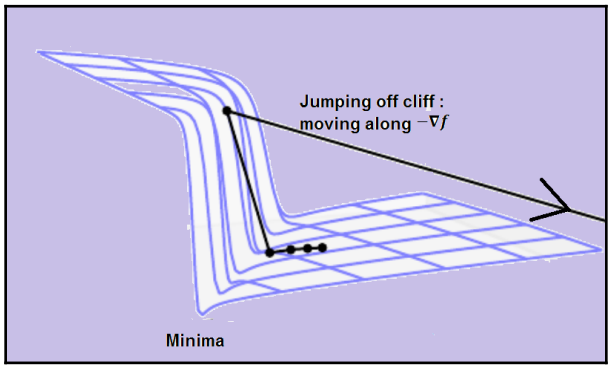

X1
X2
∫

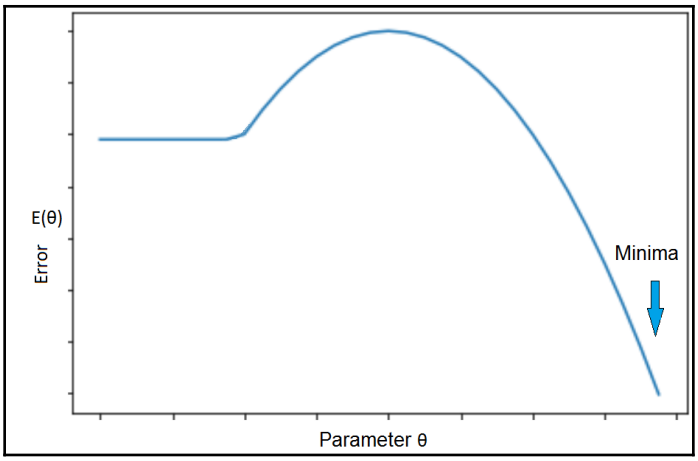


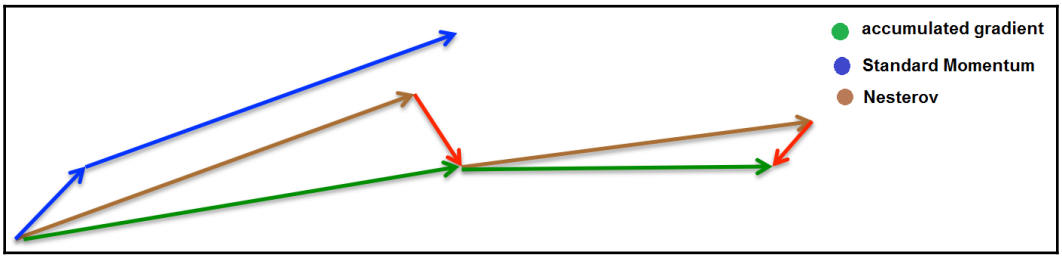
Output Of Learned XOR Function

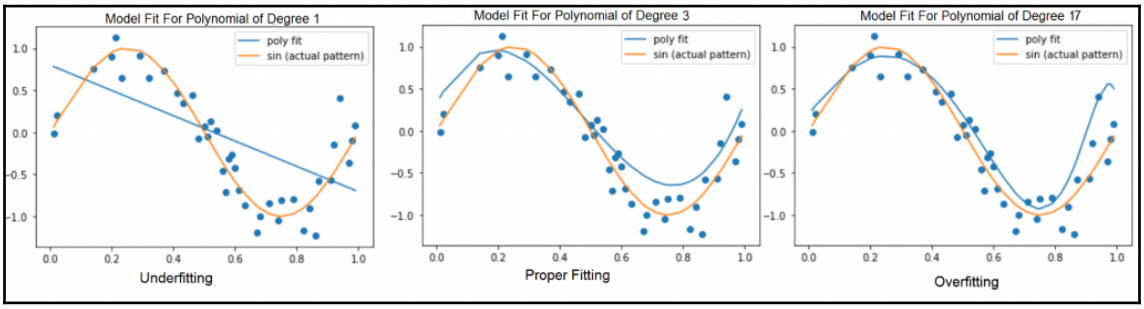
	Predictions	Y_actual
0	[0.06]	[0]
1	[0.986]	[1]
2	[0.986]	[1]
3	[0.01]	[0]

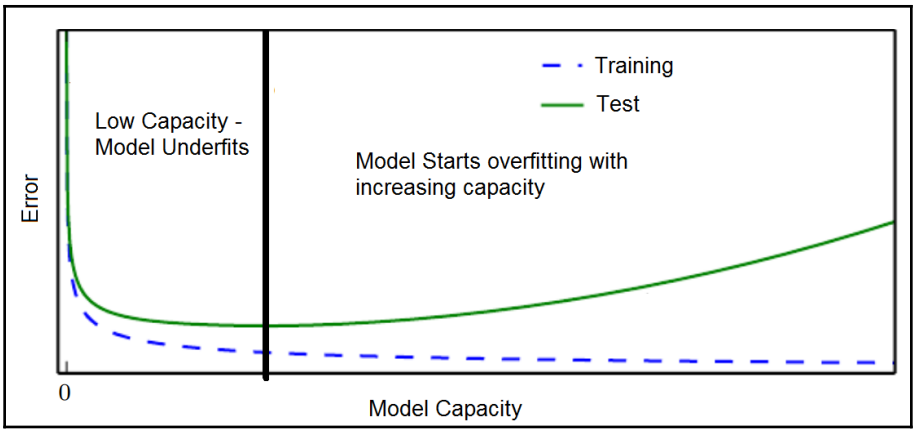


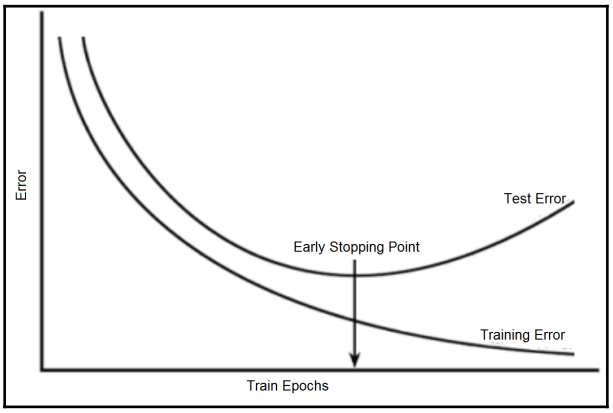



E(θ)
Error
Minima
Parameter θ

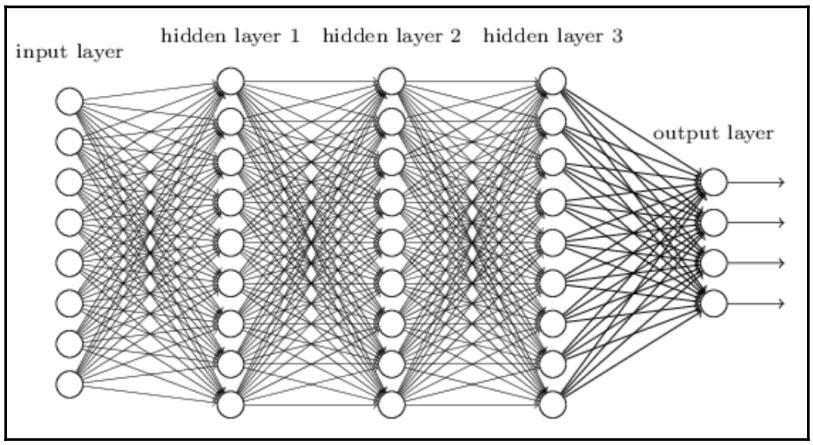

accumulated gradient
Standard Momentum
Nesterov

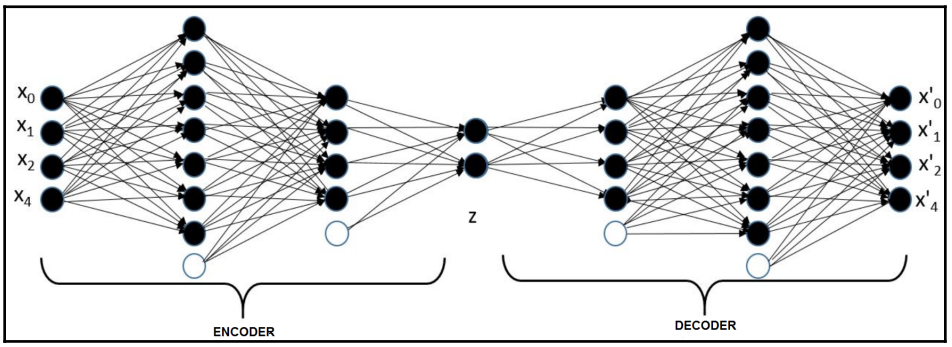

Model Fit For Polynomial of Degree 1
poly fit
sin (actual pattern)
Underfitting
Model Fit For Polynomial of Degree 3
poly fit
sin (actual pattern)
Proper Fitting
Model Fit For Polynomial of Degree 17
poly fit
sin (actual pattern)
Overfitting

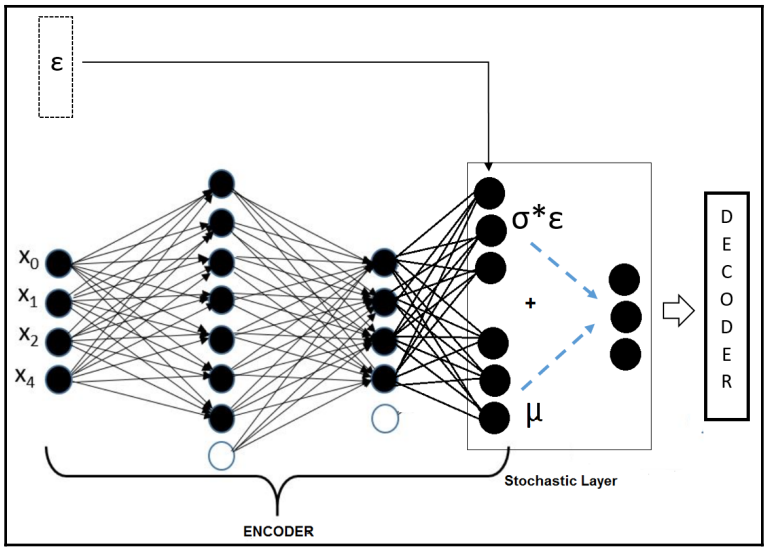

Training
Test
Low Capacity -
Model Underfits
Model Starts overfitting with
increasing capacity
Error
0
Model Capacity

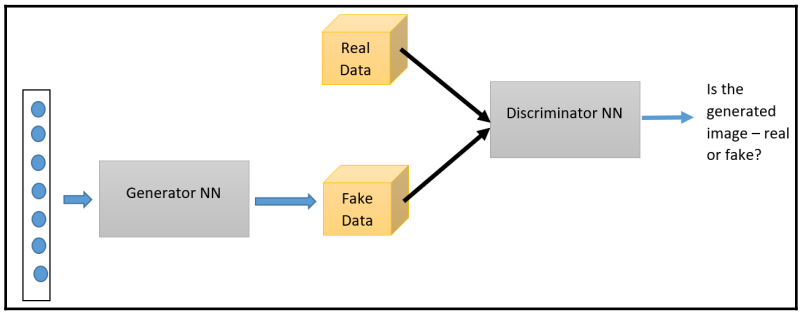

Error
Test Error
Early Stopping Point
Training Error
Train Epochs

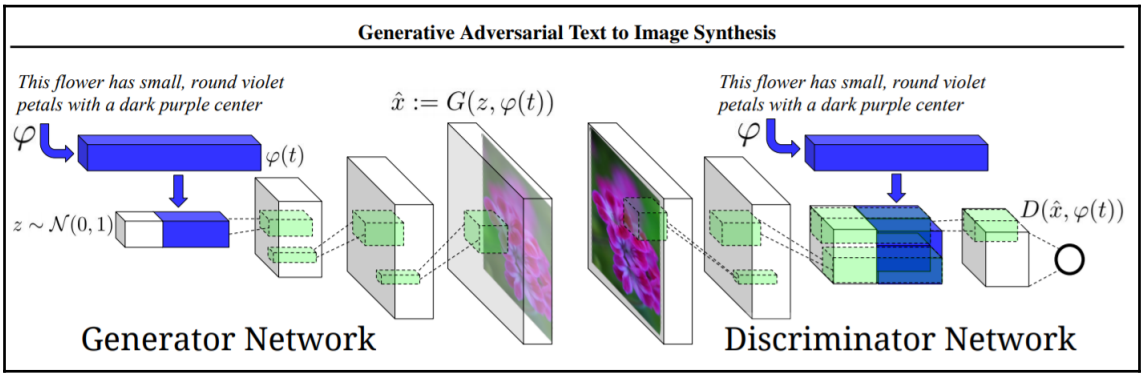
Chapter 3: Understanding Deep Learning Architectures

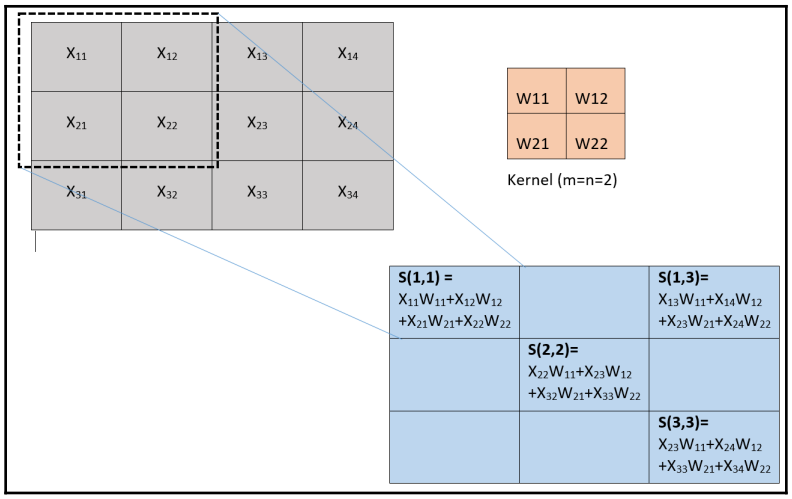


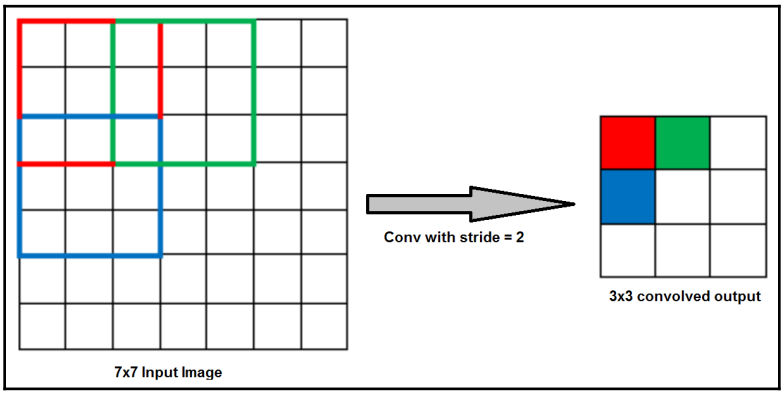


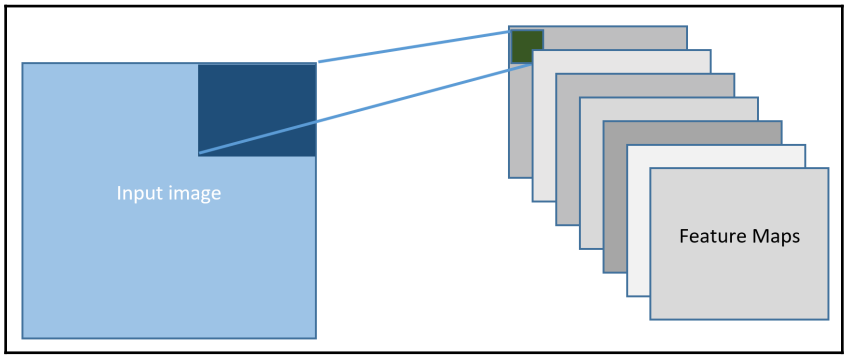

ε
x_0
x_1
x_2
x_4
σ*ε
+
μ
D E C O D E R
Stochastic Layer
ENCODER

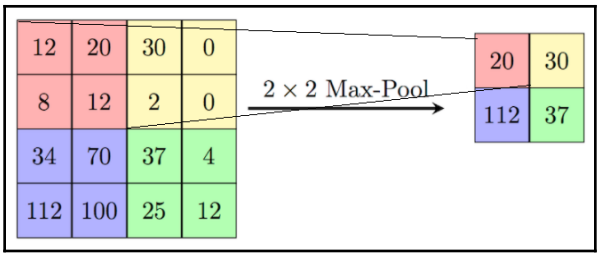

Real Data
Discriminator NN
Is the generated image – real or fake?
Generator NN
Fake Data

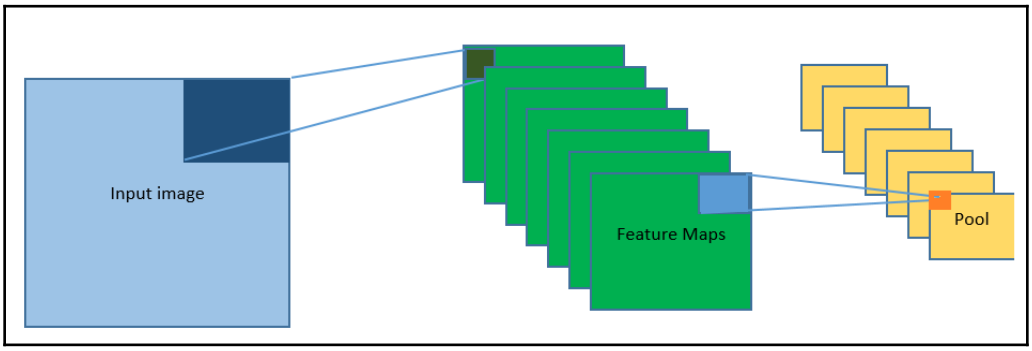

Generative Adversarial Text to Image Synthesis
This flower has small, round violet petals with a dark purple center
$\hat{x} := G(z, \varphi(t))$
This flower has small, round violet petals with a dark purple center
φ
$\varphi(t)$
$z \sim \mathcal{N}(0,1)$
$D(\hat{x}, \varphi(t))$
Generator Network
Discriminator Network

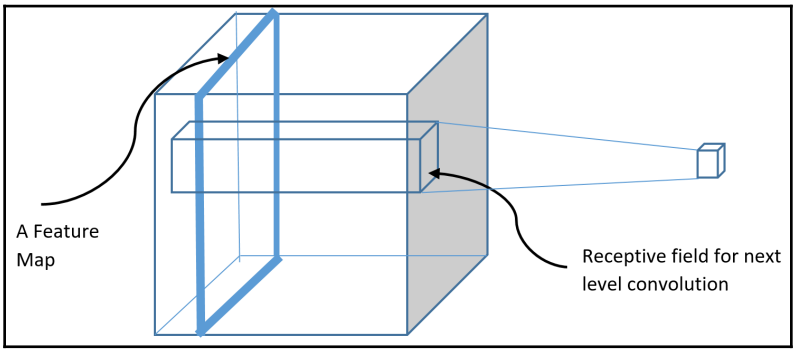

X11 X12 X13 X14
X21 X22 X23 X24
X31 X32 X33 X34
W11 W12
W21 W22
Kernel (m=n=2)
S(1,1) = X11W11+X12W12 +X21W21+X22W22
S(1,3)= X13W11+X14W12 +X23W21+X24W22
S(2,2)= X22W11+X23W12 +X32W21+X33W22
S(3,3)= X23W11+X24W12 +X33W21+X34W22

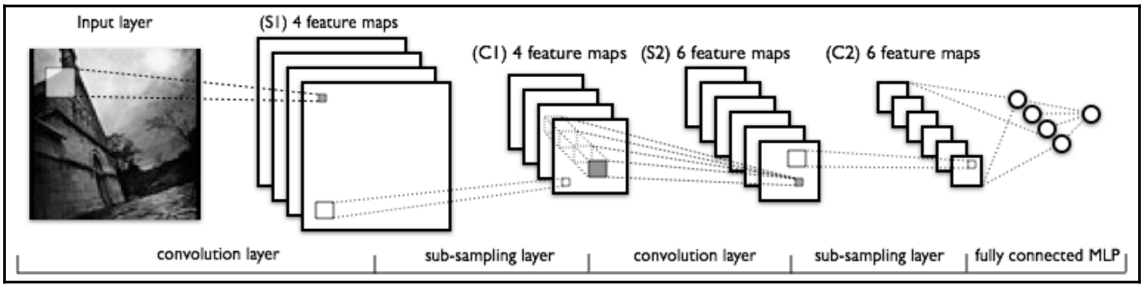

7x7 Input Image
Conv with stride = 2
3x3 convolved output


Input image
Feature Maps


12 20 30 0
8 12 2 0
34 70 37 4
112 100 25 12
2 × 2 Max-Pool
20 30
112 37


Input image
Feature Maps
Pool


A Feature Map
Receptive field for next level convolution


Input layer
(S1) 4 feature maps
(C1) 4 feature maps
(S2) 6 feature maps
(C2) 6 feature maps
convolution layer
sub-sampling layer
convolution layer
sub-sampling layer
fully connected MLP

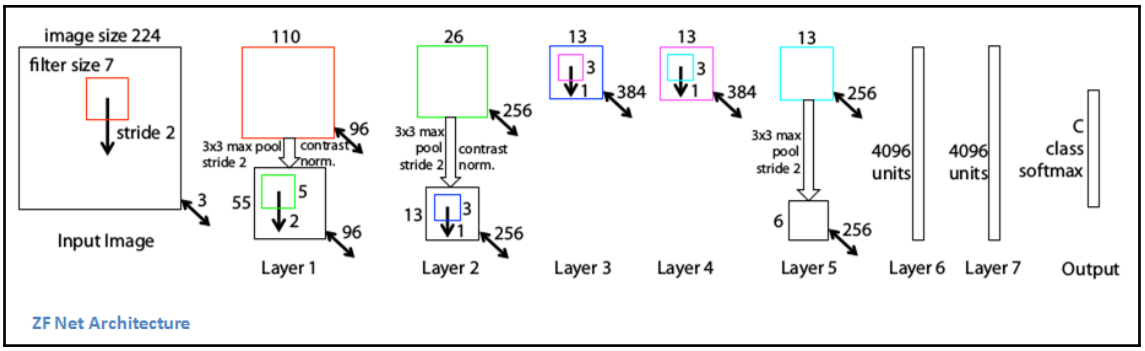
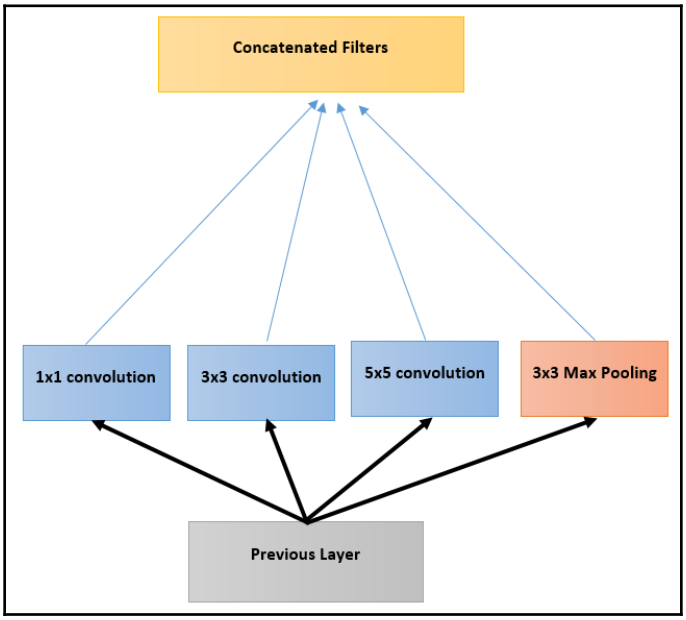
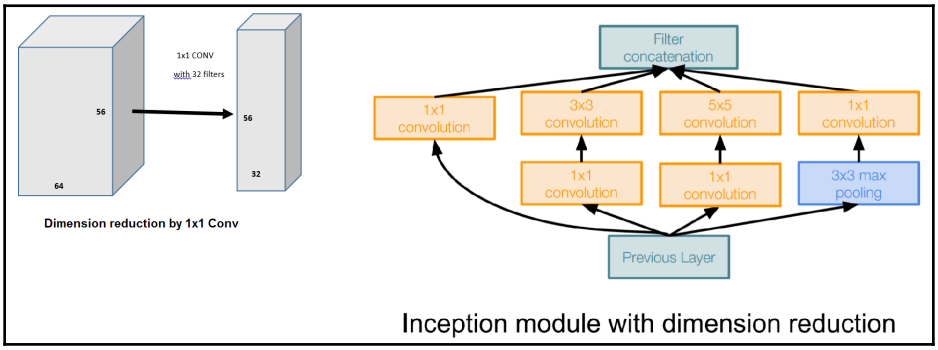
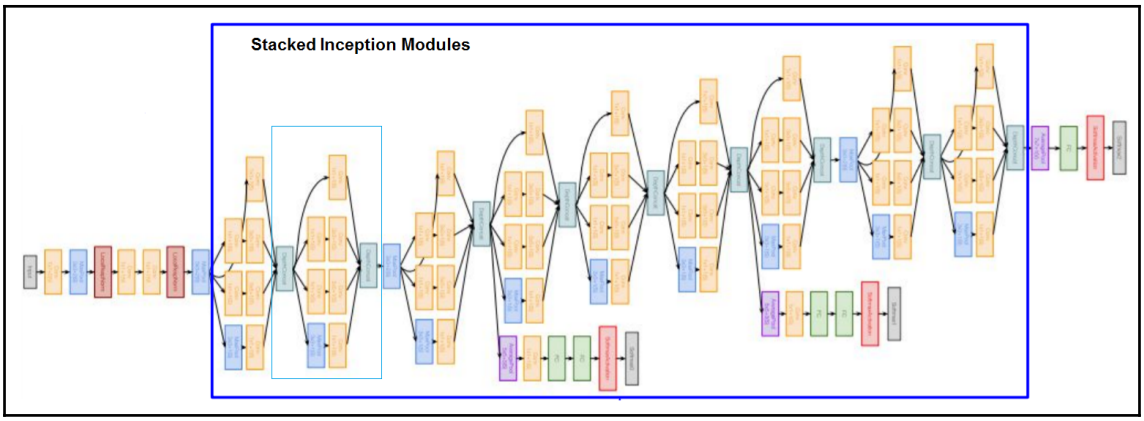
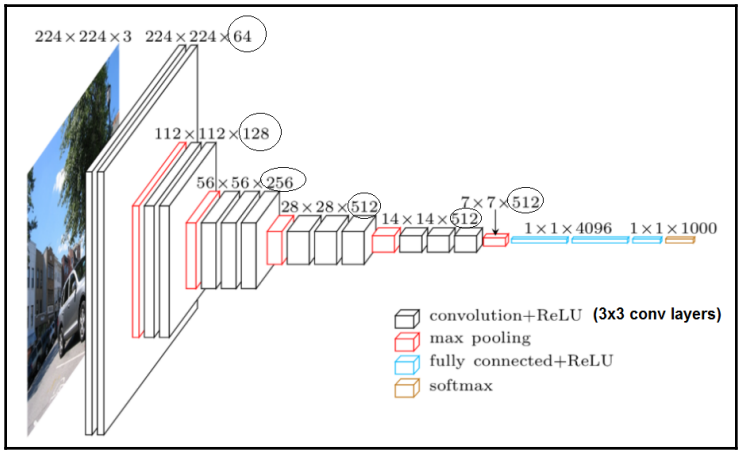


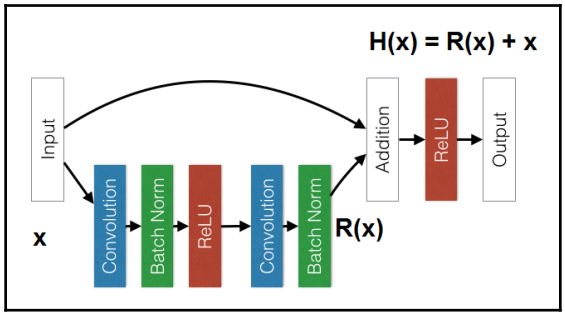
image size 224
filter size 7
stride 2
3
Input Image
110
96
3x3 max pool
stride 2
contrast
norm.
55
5
2
96
Layer 1
26
256
3x3 max
pool
stride 2
contrast
norm.
13
3
1
256
Layer 2
13
3
1
384
Layer 3
13
3
1
384
Layer 4
13
256
3x3 max
pool
stride 2
6
256
Layer 5
4096
units
Layer 6
4096
units
Layer 7
C
class
softmax
Output
ZF Net Architecture

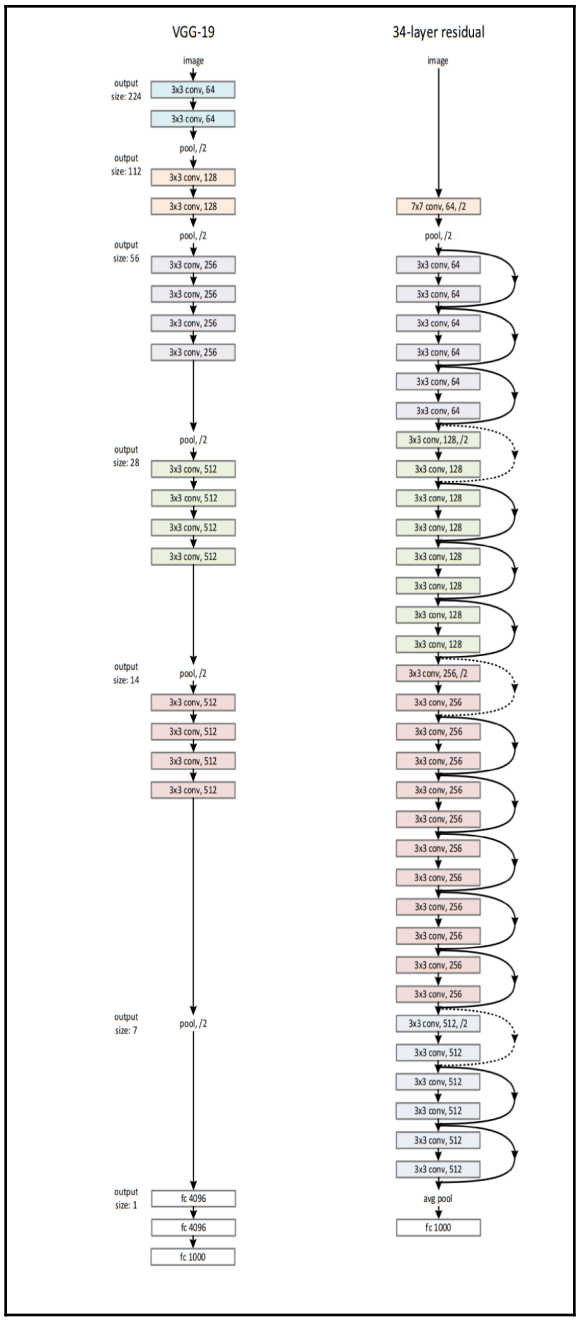

Concatenated Filters
1x1 convolution
3x3 convolution
5x5 convolution
3x3 Max Pooling
Previous Layer

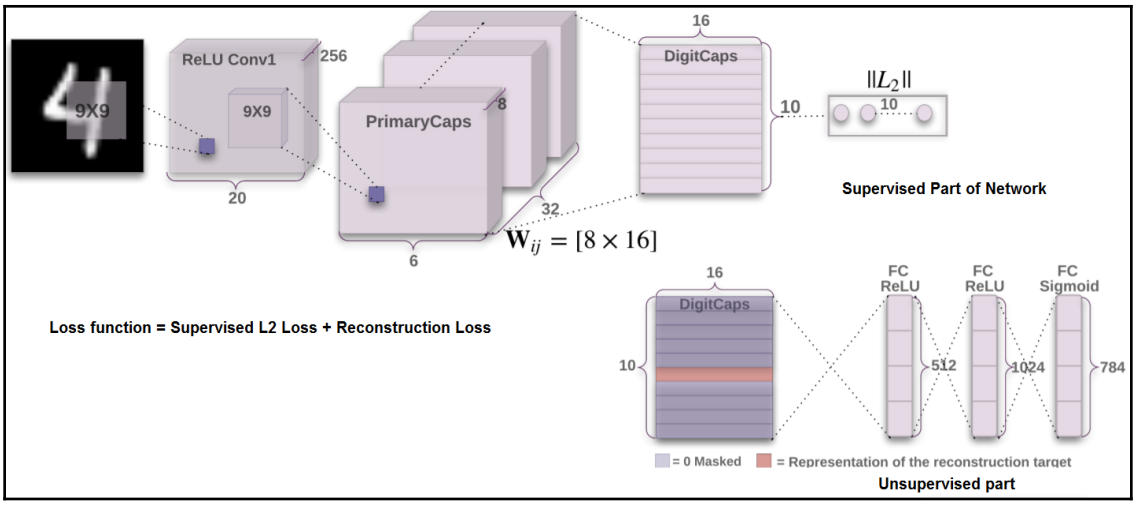

1x1 CONV
with 32 filters
56
64
56
32
Dimension reduction by 1x1 Conv
Filter
concatenation
1x1
convolution
3x3
convolution
5x5
convolution
1x1
convolution
1x1
convolution
1x1
convolution
3x3 max
pooling
Previous Layer
Inception module with dimension reduction

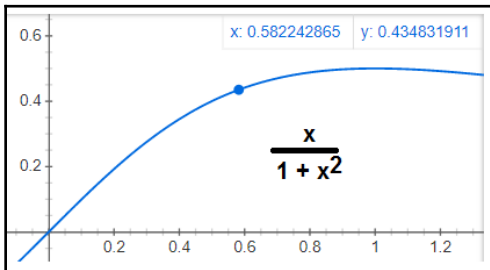

Stacked Inception Modules


224×224×3
224×224×64
112×112×128
56×56×256
28×28×512
14×14×512
7×7×512
1×1×4096
1×1×1000
convolution+ReLU (3x3 conv layers)
max pooling
fully connected+ReLU
softmax


H(x) = R(x) + x
Input
x
Convolution
Batch Norm
ReLU
Convolution
Batch Norm
R(x)
Addition
ReLU
Output

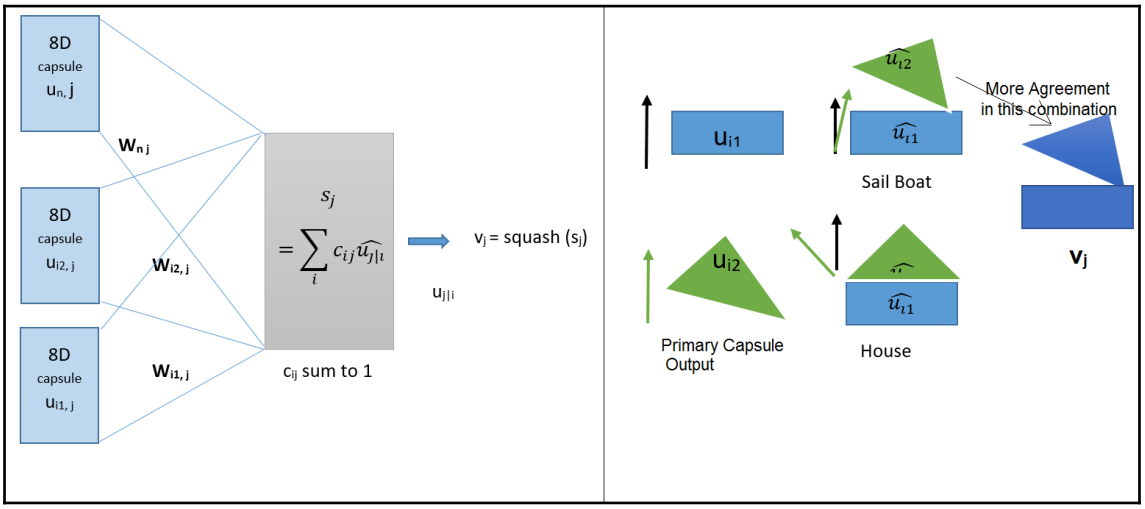

VGG-19
image
output size: 224
3x3 conv, 64
3x3 conv, 64
pool, /2
output size: 112
3x3 conv, 128
3x3 conv, 128
pool, /2
output size: 56
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool, /2
output size: 28
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
output size: 14
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
output size: 7
fc 4096
fc 4096
fc 1000
output size: 1
34-layer residual
image
7x7 conv, 64, /2
pool, /2
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, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
avg pool
fc 1000

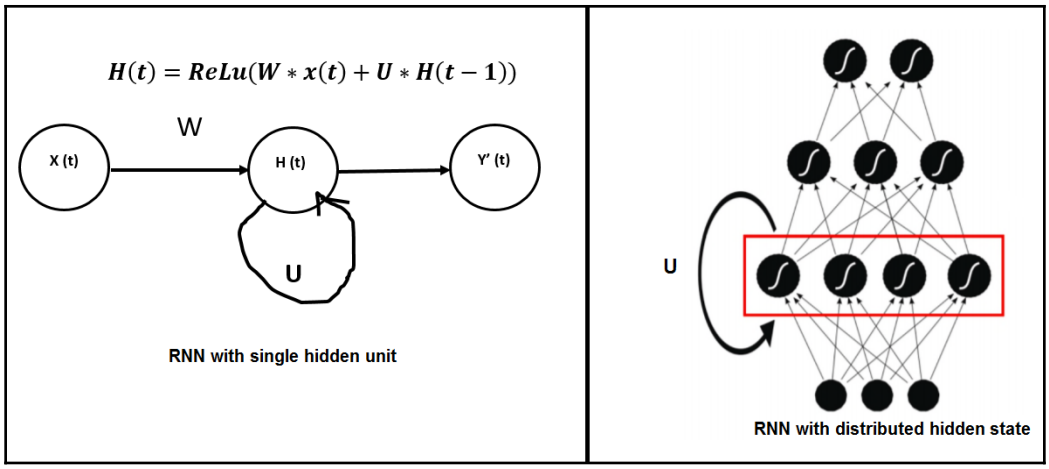


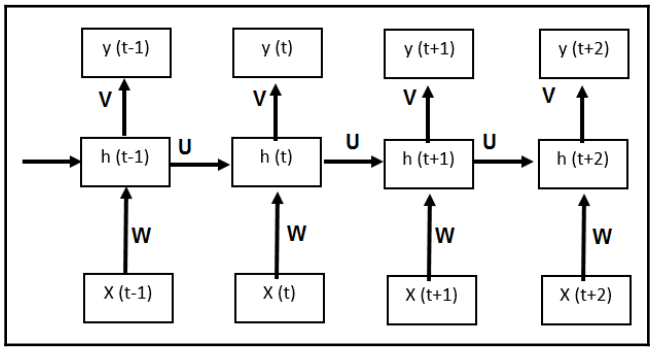


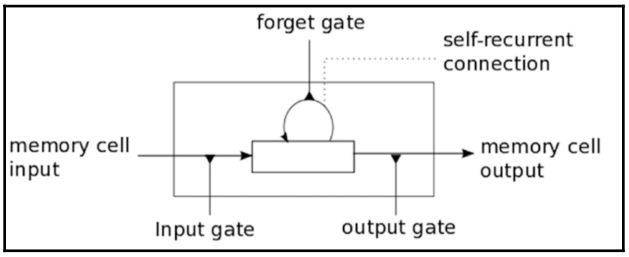
$$\mathbf{v}_j = \frac{||\mathbf{s}_j||^2}{1+||\mathbf{s}_j||^2} \frac{\mathbf{s}_j}{||\mathbf{s}_j||}$$

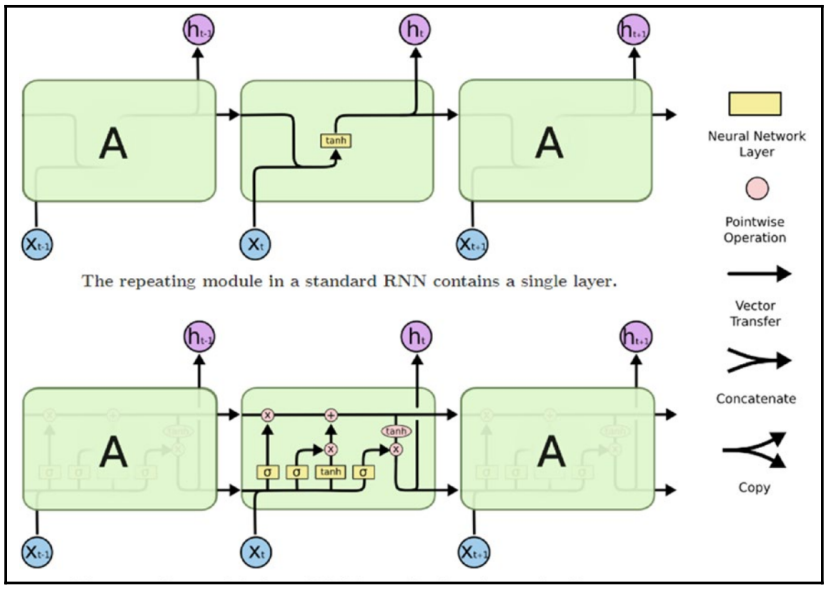
v_j is the vector output of capsule j and s_j is its total input



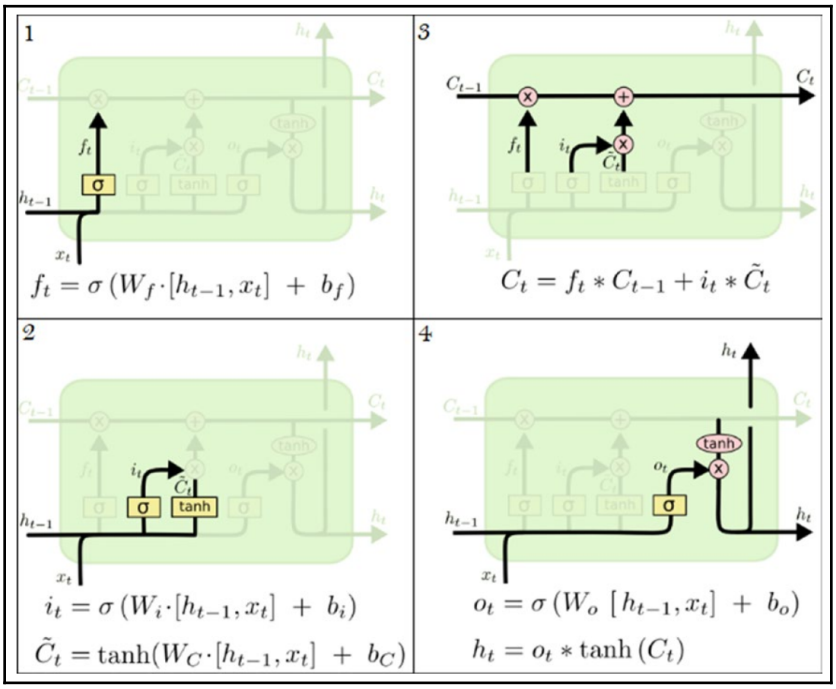

$H(t) = ReLu(W * x(t) + U * H(t-1))$
W
X (t)
H (t)
Y' (t)
U
RNN with single hidden unit
U
RNN with distributed hidden state

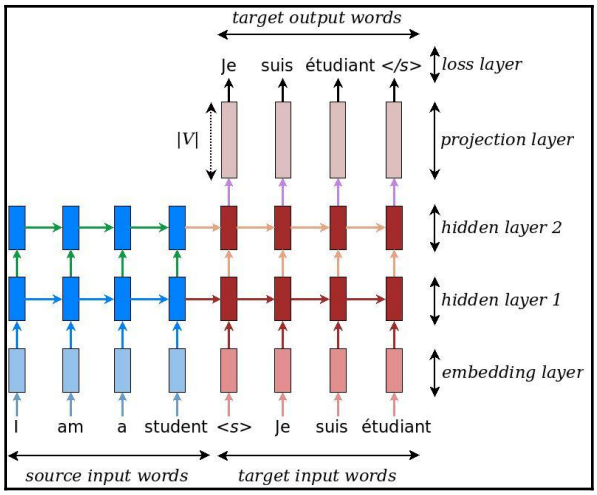

y (t-1)
y (t)
y (t+1)
y (t+2)
V
U
h (t-1)
h (t)
h (t+1)
h (t+2)
W
X (t-1)
X (t)
X (t+1)
X (t+2)

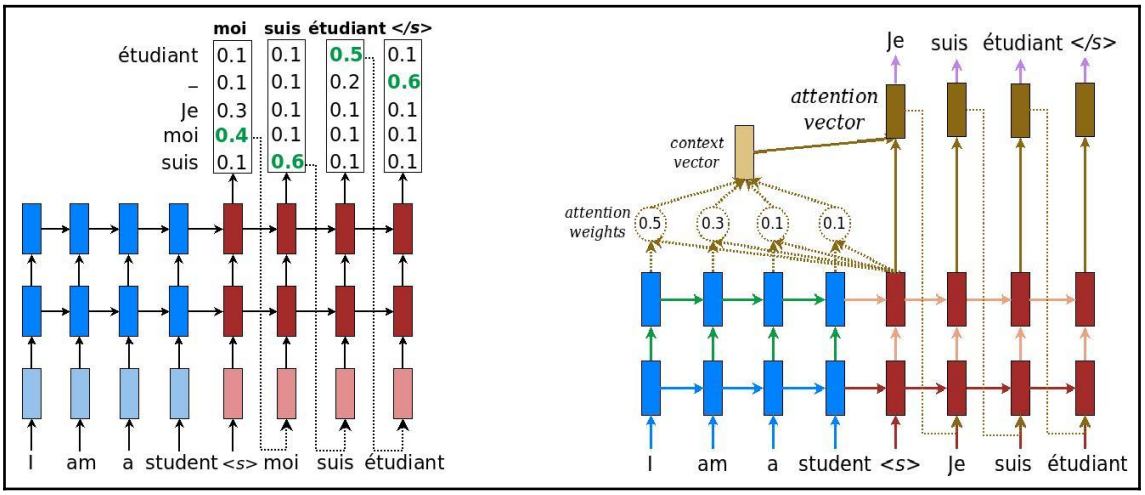

forget gate
self-recurrent connection
memory cell input
memory cell output
Input gate
output gate

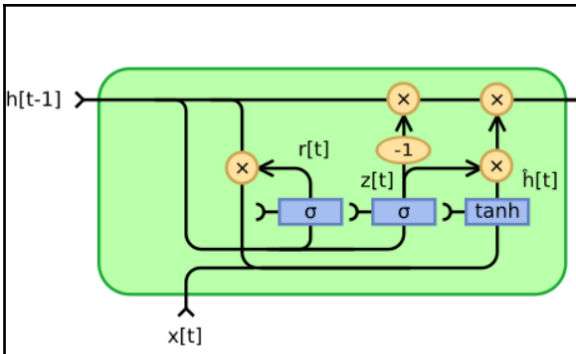


The repeating module in a standard RNN contains a single layer.




target output words
Je
suis
étudiant
</s>
loss layer
|V|
projection layer
hidden layer 2
hidden layer 1
embedding layer
I
am
a
student
<s>
Je
suis
étudiant
source input words
target input words


moi
suis
étudiant
</s>
étudiant
_
Je
moi
suis
0.1
0.1
0.3
0.4
0.1
0.1
0.1
0.1
0.1
0.6
0.5
0.2
0.1
0.1
0.1
0.6
0.1
0.1
0.1
0.1
0.1
I
am
a
student
<s>
moi
suis
étudiant
Je
suis
étudiant
</s>
attention
vector
context
vector
attention
weights
0.5
0.3
0.1
0.1
I
am
a
student
<s>
Je
suis
étudiant

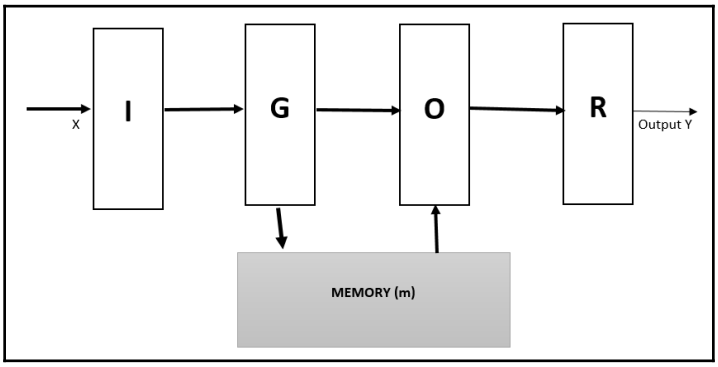


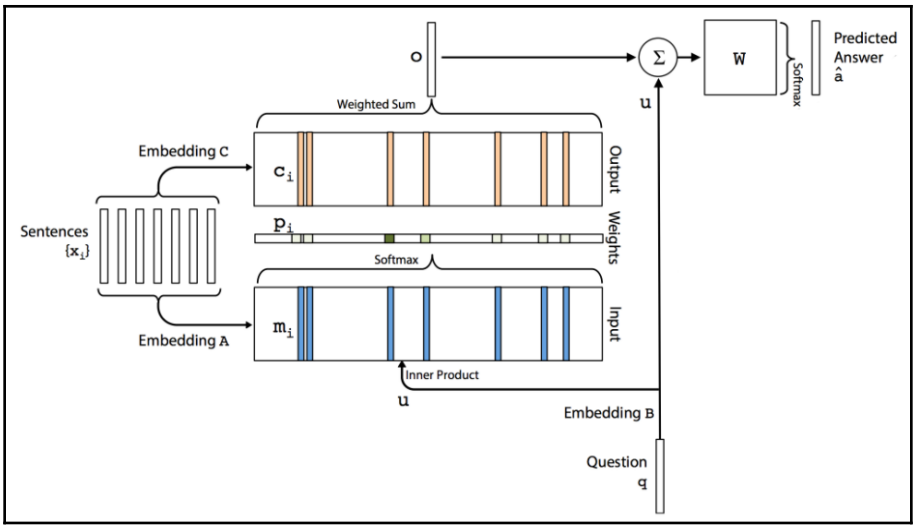
Initially, for $t = 0$, the output vector is $h_0 = 0$.

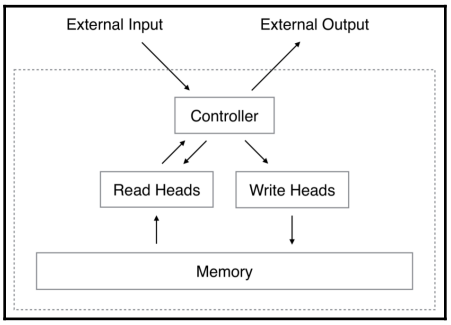
$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z)$$
$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r)$$
$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h)$$

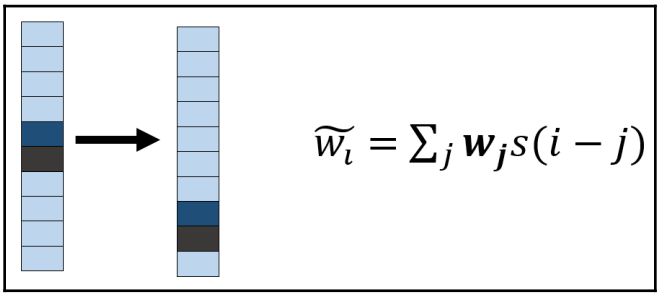
Variables

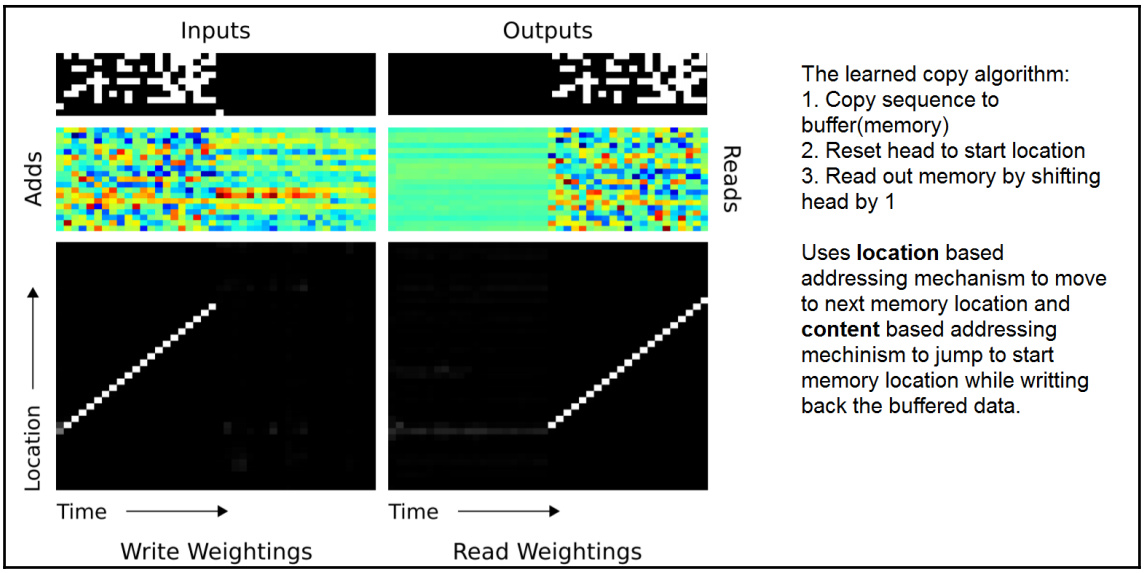
- x_t: input vector
- h_t: output vector
- z_t: update gate vector
- r_t: reset gate vector
- W, U and b: parameter matrices and vector




o
Σ
W
Softmax
Predicted Answer
â
u
Weighted Sum
Embedding C
c_i
Output
p_i
Weights
Softmax
Sentences {x_i}
m_i
Input
Embedding A
Inner Product
u
Embedding B
Question q

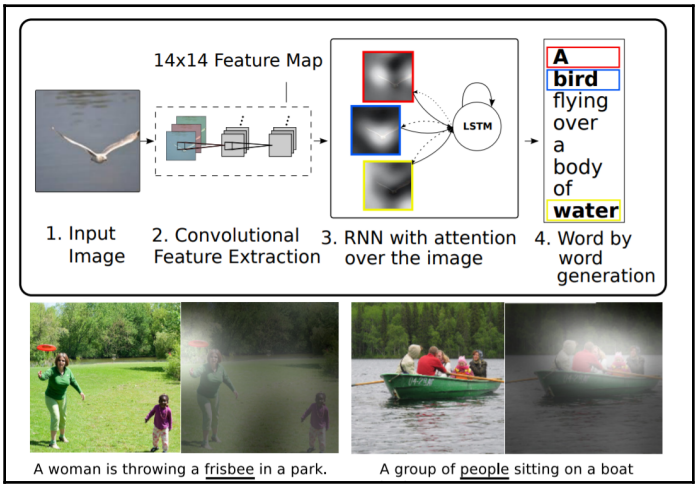

External Input
External Output
Controller
Read Heads
Write Heads
Memory


$\widetilde{w_i} = \sum_j \boldsymbol{w_j} s(i-j)$

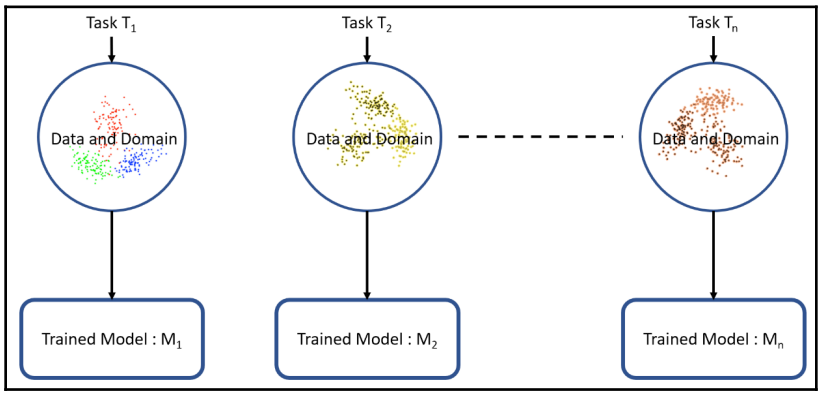


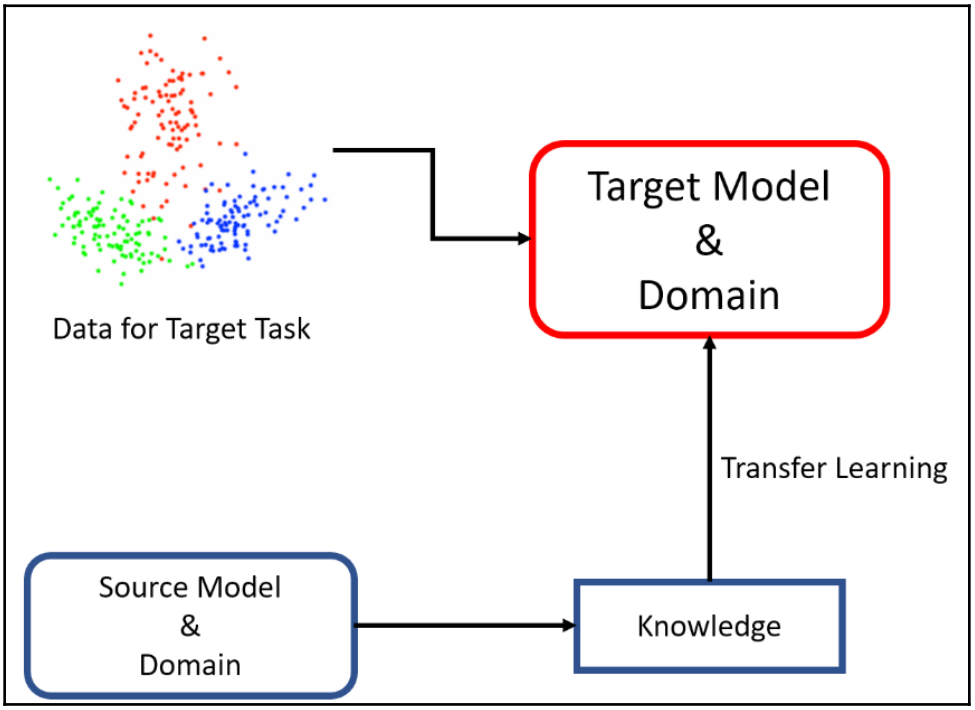
The learned copy algorithm:
1. Copy sequence to buffer(memory)
2. Reset head to start location
3. Read out memory by shifting head by 1

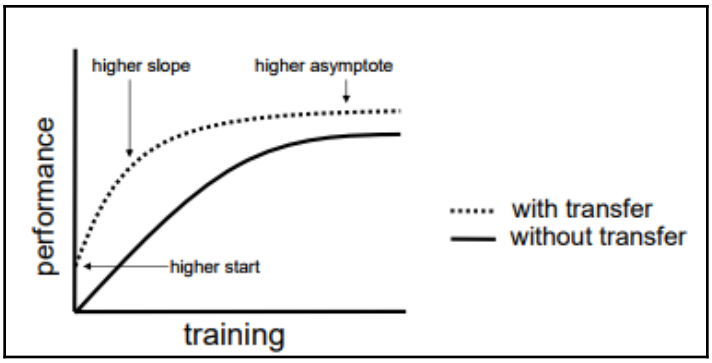
Uses **location** based addressing mechanism to move to next memory location and **content** based addressing mechinism to jump to start memory location while writting back the buffered data.

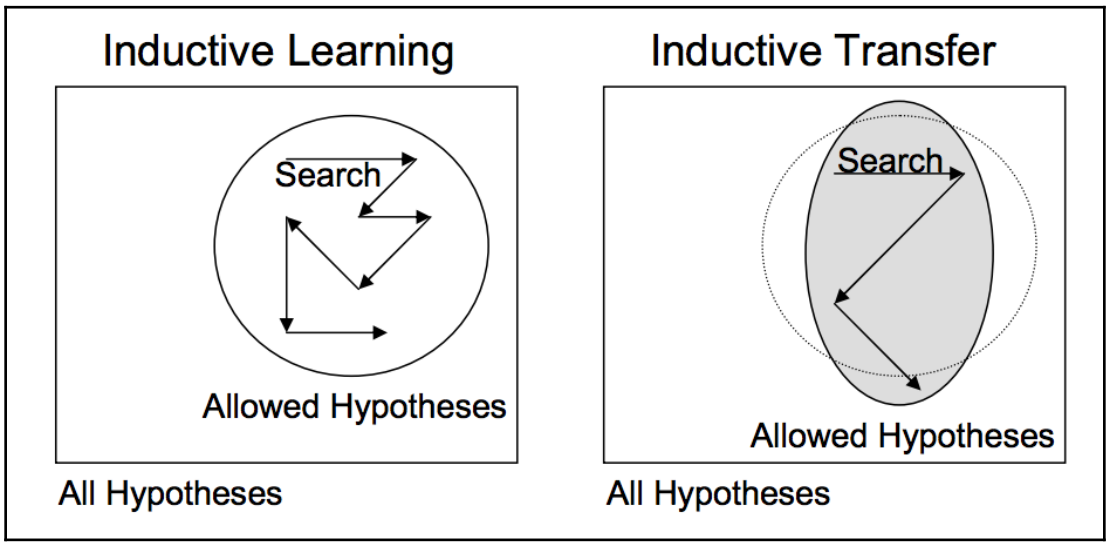


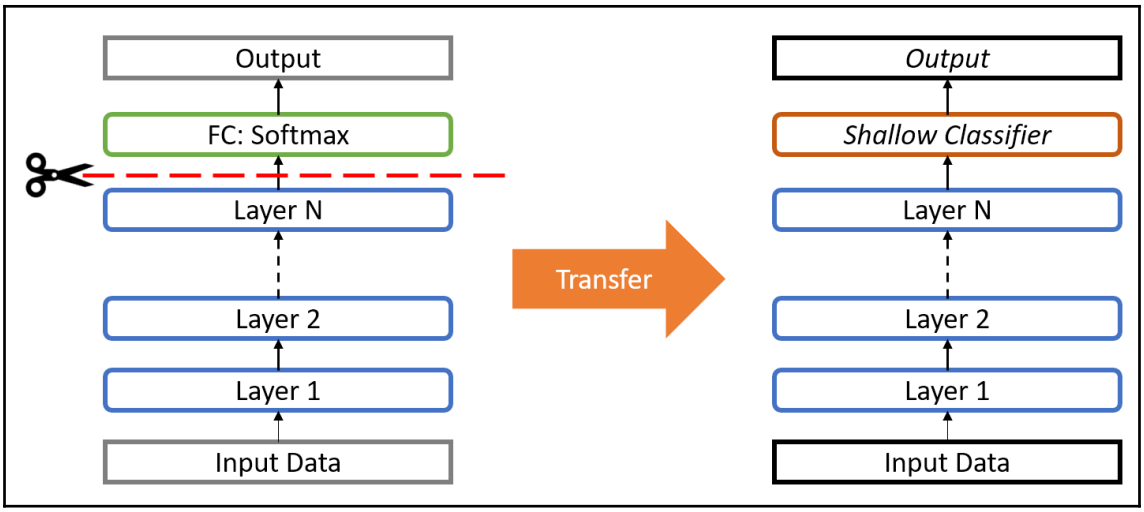
Chapter 4: Transfer Learning Fundamentals

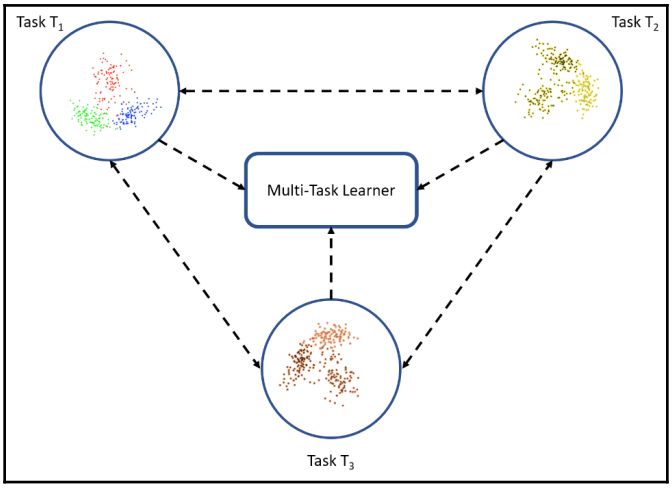




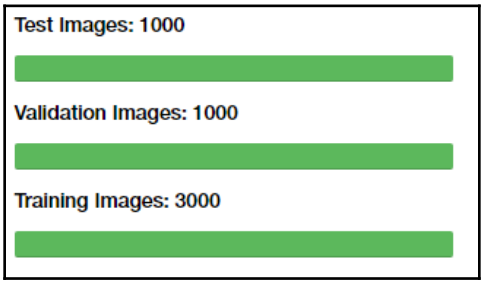

higher slope
higher asymptote
performance
higher start
training
with transfer
without transfer

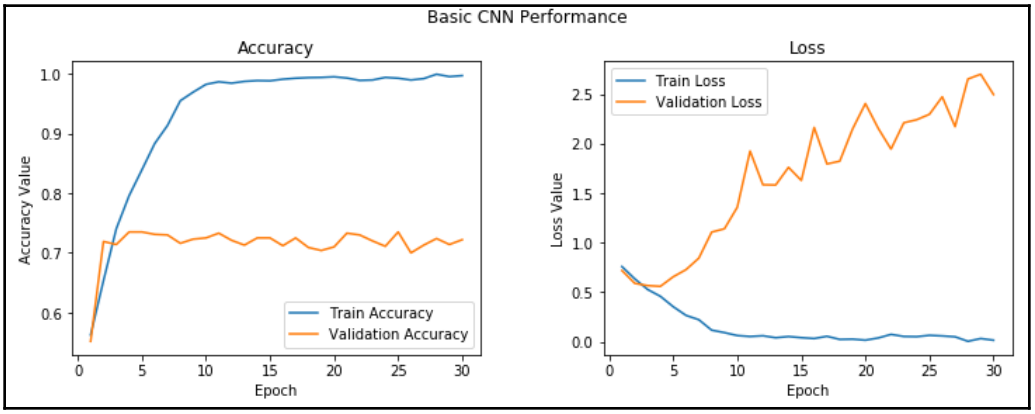

Inductive Learning
Search
Allowed Hypotheses
All Hypotheses
Inductive Transfer
Search
Allowed Hypotheses
All Hypotheses

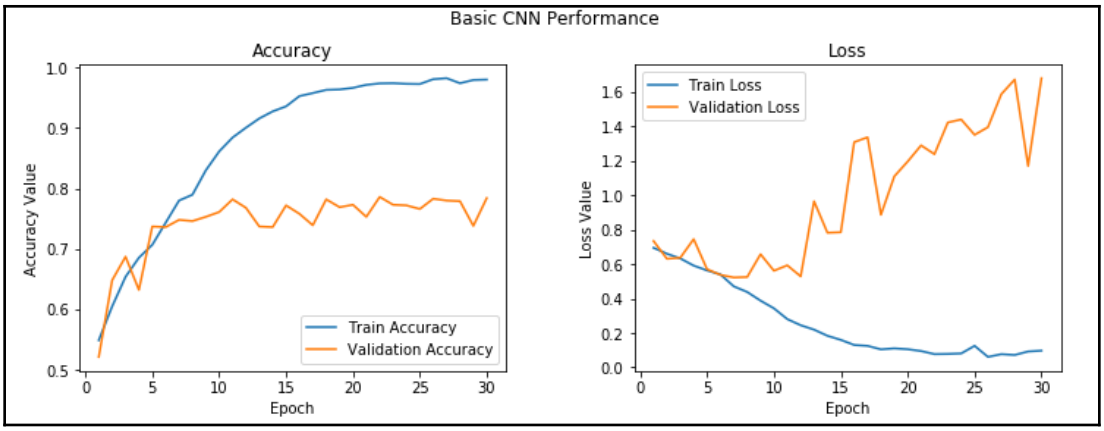

Output
FC: Softmax
Layer N
Layer 2
Layer 1
Input Data
Transfer
Output
Shallow Classifier
Layer N
Layer 2
Layer 1
Input Data

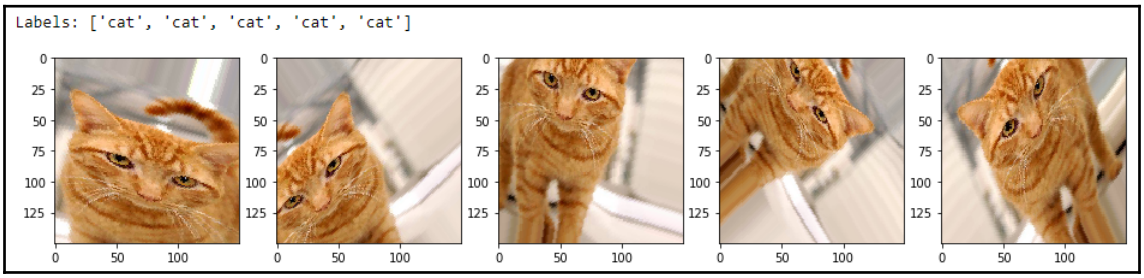

Task T_1
Task T_2
Multi-Task Learner
Task T_3

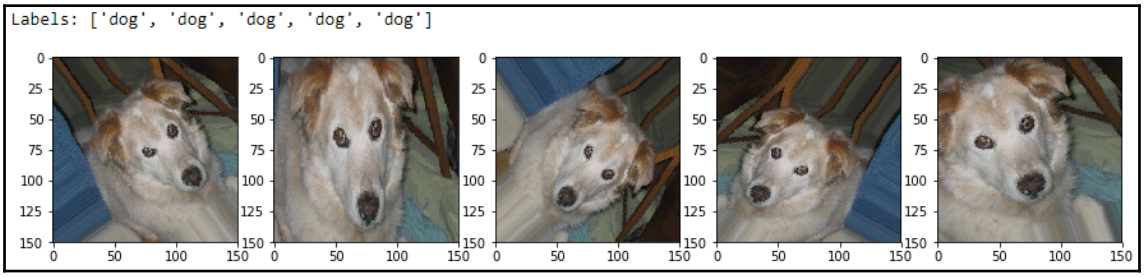
Chapter 5: Unleashing the Power of Transfer Learning

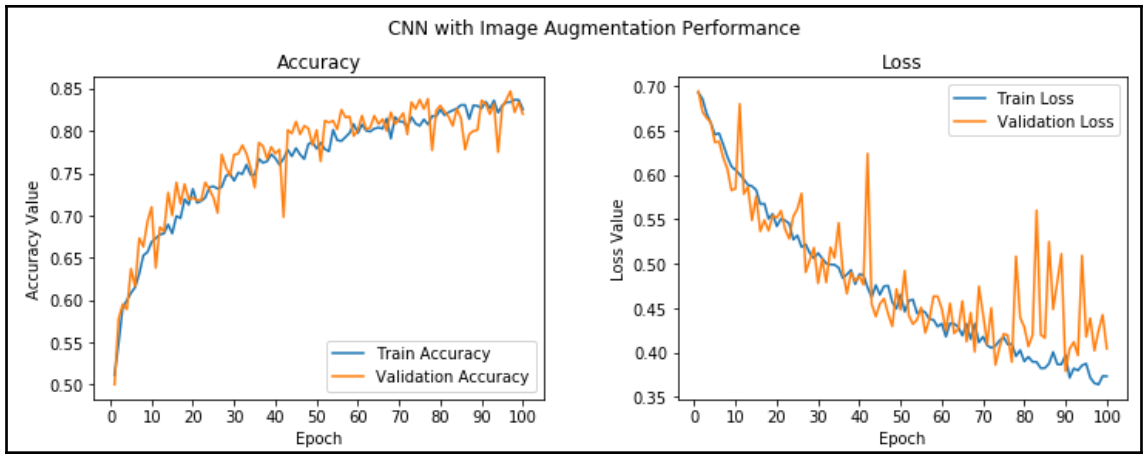


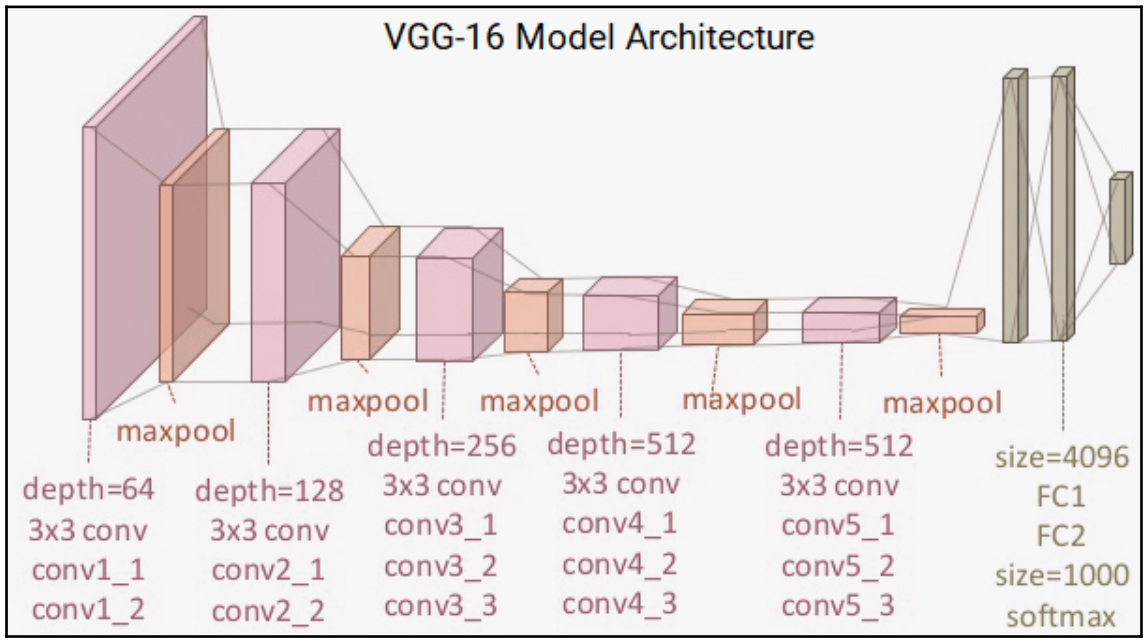


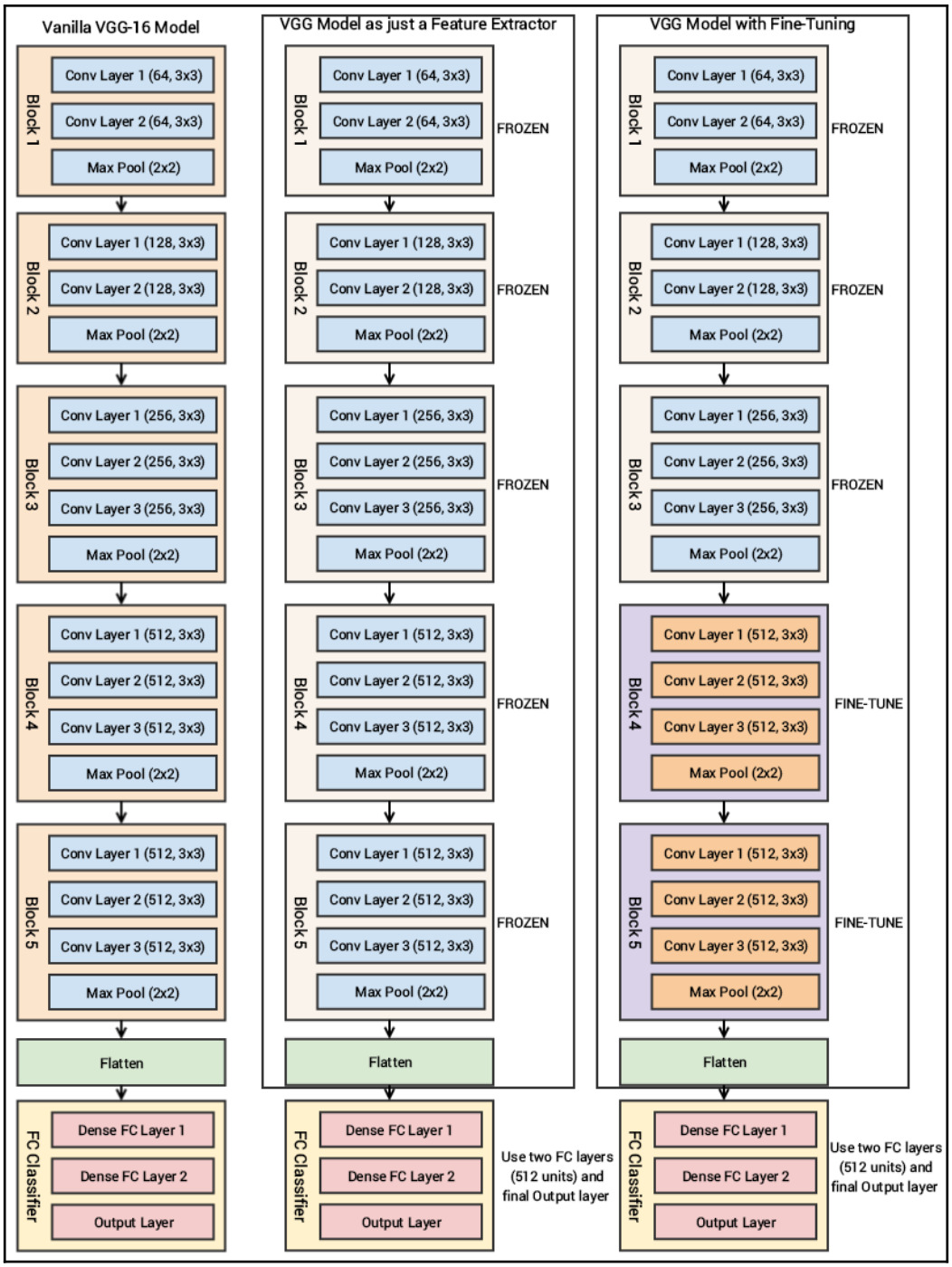

Basic CNN Performance
Accuracy
Accuracy Value
Epoch
Train Accuracy
Validation Accuracy
Loss
Loss Value
Epoch
Train Loss
Validation Loss


Labels: ['cat', 'cat', 'cat', 'cat', 'cat']

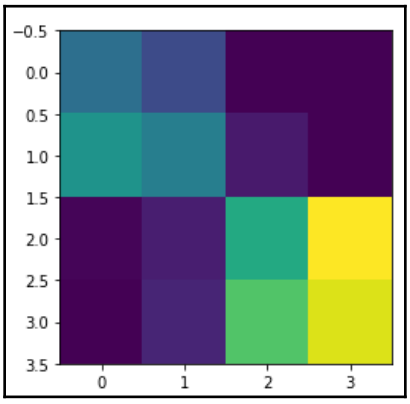

Labels: ['dog', 'dog', 'dog', 'dog', 'dog']

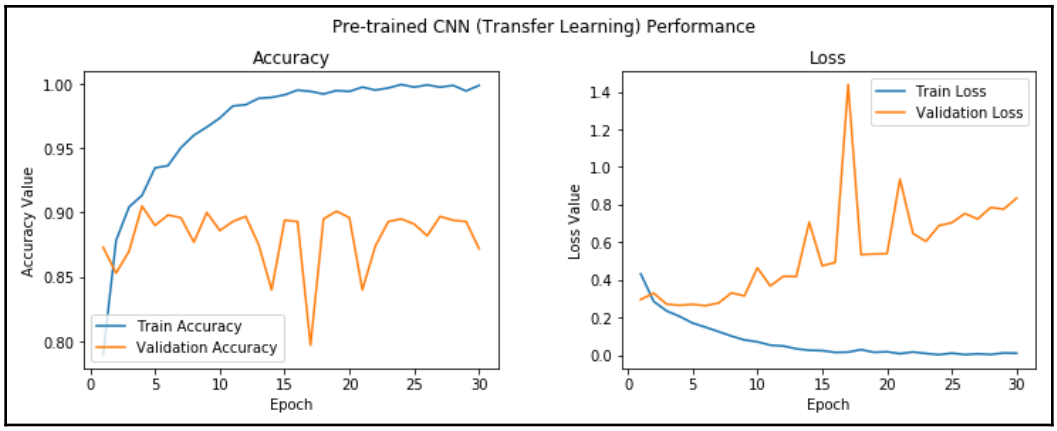

CNN with Image Augmentation Performance
Accuracy
Accuracy Value
Epoch
Train Accuracy
Validation Accuracy
Loss
Loss Value
Epoch
Train Loss
Validation Loss

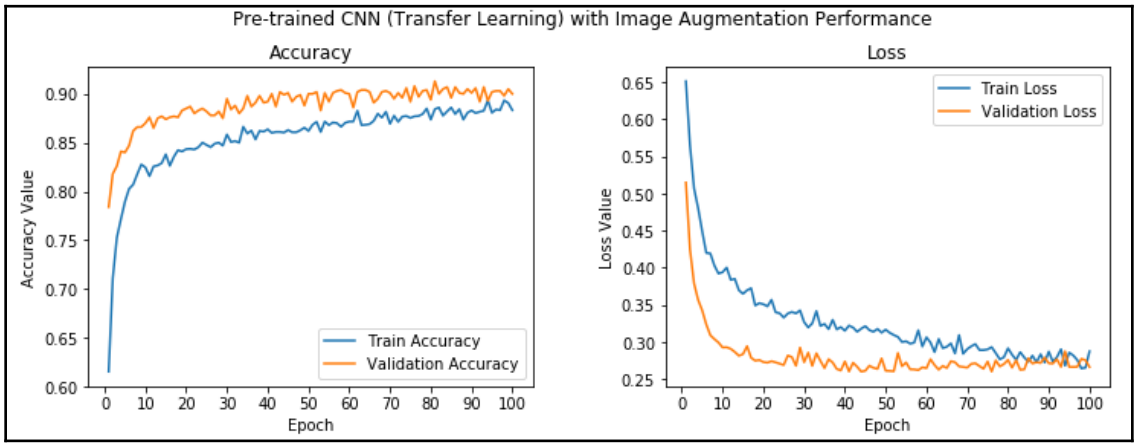

VGG-16 Model Architecture
maxpool
maxpool
maxpool
maxpool
maxpool
maxpool
depth=64
3x3 conv
conv1_1
conv1_2
depth=128
3x3 conv
conv2_1
conv2_2
depth=256
3x3 conv
conv3_1
conv3_2
conv3_3
depth=512
3x3 conv
conv4_1
conv4_2
conv4_3
depth=512
3x3 conv
conv5_1
conv5_2
conv5_3
size=4096
FC1
FC2
size=1000
softmax


Vanilla VGG-16 Model
VGG Model as just a Feature Extractor
VGG Model with Fine-Tuning
Block 1
Conv Layer 1 (64, 3x3)
Conv Layer 2 (64, 3x3)
Max Pool (2x2)
Block 2
Conv Layer 1 (128, 3x3)
Conv Layer 2 (128, 3x3)
Max Pool (2x2)
Block 3
Conv Layer 1 (256, 3x3)
Conv Layer 2 (256, 3x3)
Conv Layer 3 (256, 3x3)
Max Pool (2x2)
Block 4
Conv Layer 1 (512, 3x3)
Conv Layer 2 (512, 3x3)
Conv Layer 3 (512, 3x3)
Max Pool (2x2)
Block 5
Conv Layer 1 (512, 3x3)
Conv Layer 2 (512, 3x3)
Conv Layer 3 (512, 3x3)
Max Pool (2x2)
FROZEN
FINE-TUNE
Flatten
FC Classifier
Dense FC Layer 1
Dense FC Layer 2
Output Layer
Use two FC layers (512 units) and final Output layer

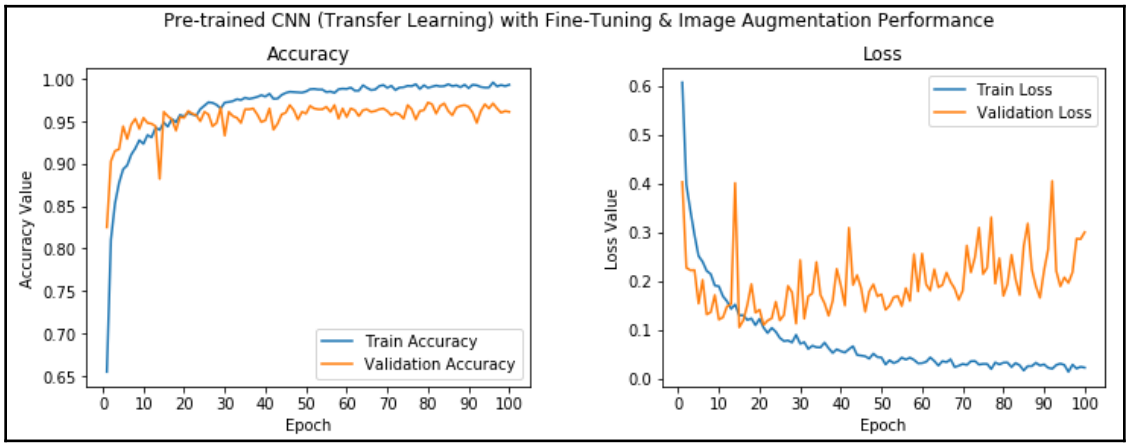
	Layer Type	Layer Name	Layer Trainable
0	<keras.engine.topology.InputLayer object at 0x7f26c86b2518>	input_1	False
1	<keras.layers.convolutional.Conv2D object at 0x7f277c9fc080>	block1_conv1	False
2	<keras.layers.convolutional.Conv2D object at 0x7f26c86b26d8>	block1_conv2	False
3	<keras.layers.pooling.MaxPooling2D object at 0x7f26c86e6c88>	block1_pool	False
4	<keras.layers.convolutional.Conv2D object at 0x7f26c867dc18>	block2_conv1	False
5	<keras.layers.convolutional.Conv2D object at 0x7f26c8690f28>	block2_conv2	False
6	<keras.layers.pooling.MaxPooling2D object at 0x7f26c869e5c0>	block2_pool	False
7	<keras.layers.convolutional.Conv2D object at 0x7f26c863f828>	block3_conv1	False
8	<keras.layers.convolutional.Conv2D object at 0x7f26c863f128>	block3_conv2	False
9	<keras.layers.convolutional.Conv2D object at 0x7f26c86607b8>	block3_conv3	False
10	<keras.layers.pooling.MaxPooling2D object at 0x7f26c83d7d68>	block3_pool	False
11	<keras.layers.convolutional.Conv2D object at 0x7f26c83fd358>	block4_conv1	False
12	<keras.layers.convolutional.Conv2D object at 0x7f26c83fddd8>	block4_conv2	False
13	<keras.layers.convolutional.Conv2D object at 0x7f26c839da20>	block4_conv3	False
14	<keras.layers.pooling.MaxPooling2D object at 0x7f26c83ac1d0>	block4_pool	False
15	<keras.layers.convolutional.Conv2D object at 0x7f26c834e978>	block5_conv1	False
16	<keras.layers.convolutional.Conv2D object at 0x7f271a15eb38>	block5_conv2	False
17	<keras.layers.convolutional.Conv2D object at 0x7f26c8371d68>	block5_conv3	False
18	<keras.layers.pooling.MaxPooling2D object at 0x7f26c8314b00>	block5_pool	False
19	<keras.layers.core.Flatten object at 0x7f26c828bda0>	flatten_1	False

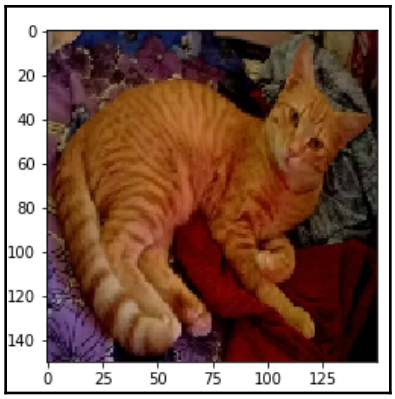


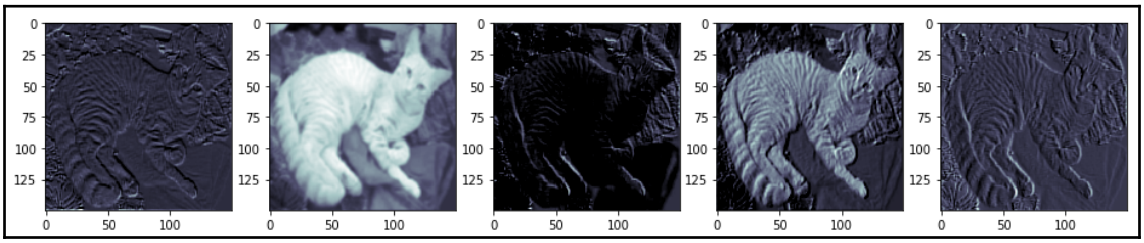

Pre-trained CNN (Transfer Learning) Performance
Accuracy
Accuracy Value
Epoch
Train Accuracy
Validation Accuracy
Loss
Loss Value
Epoch
Train Loss
Validation Loss

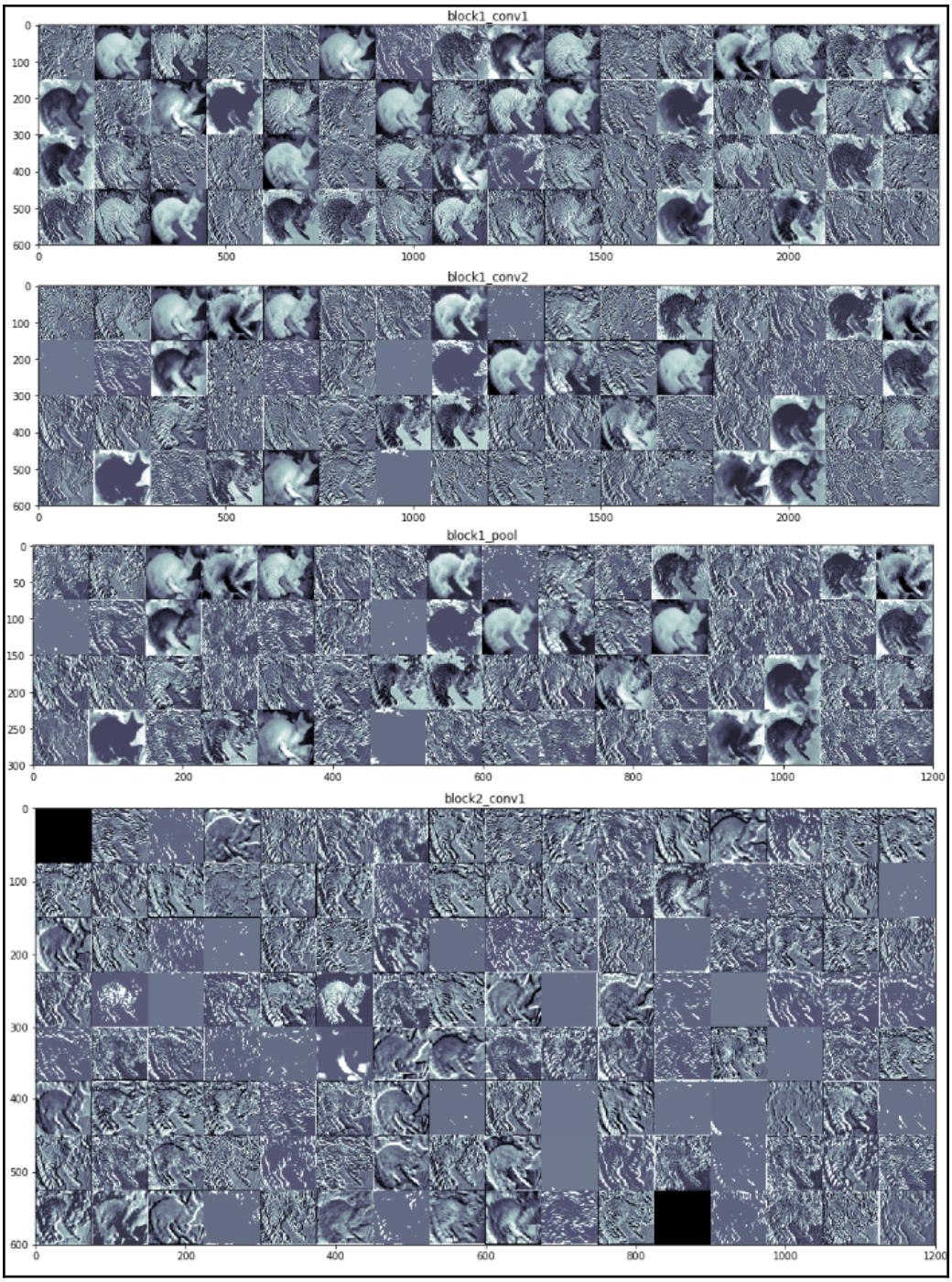

Pre-trained CNN (Transfer Learning) with Image Augmentation Performance
Accuracy
Accuracy Value
Epoch
Train Accuracy
Validation Accuracy
Loss
Loss Value
Epoch
Train Loss
Validation Loss

	Layer Type	Layer Name	Layer Trainable
0	<keras.engine.topology.InputLayer object at 0x7f26c86b2518>	input_1	False
1	<keras.layers.convolutional.Conv2D object at 0x7f277c9fc080>	block1_conv1	False
2	<keras.layers.convolutional.Conv2D object at 0x7f26c86b26d8>	block1_conv2	False
3	<keras.layers.pooling.MaxPooling2D object at 0x7f26c86e6c88>	block1_pool	False
4	<keras.layers.convolutional.Conv2D object at 0x7f26c867dc18>	block2_conv1	False
5	<keras.layers.convolutional.Conv2D object at 0x7f26c8690f28>	block2_conv2	False
6	<keras.layers.pooling.MaxPooling2D object at 0x7f26c869e5c0>	block2_pool	False
7	<keras.layers.convolutional.Conv2D object at 0x7f26c863f828>	block3_conv1	False
8	<keras.layers.convolutional.Conv2D object at 0x7f26c863f128>	block3_conv2	False
9	<keras.layers.convolutional.Conv2D object at 0x7f26c86607b8>	block3_conv3	False
10	<keras.layers.pooling.MaxPooling2D object at 0x7f26c83d7d68>	block3_pool	False
11	<keras.layers.convolutional.Conv2D object at 0x7f26c83fd358>	block4_conv1	True
12	<keras.layers.convolutional.Conv2D object at 0x7f26c83fddd8>	block4_conv2	True
13	<keras.layers.convolutional.Conv2D object at 0x7f26c839da20>	block4_conv3	True
14	<keras.layers.pooling.MaxPooling2D object at 0x7f26c83ac1d0>	block4_pool	True
15	<keras.layers.convolutional.Conv2D object at 0x7f26c834e978>	block5_conv1	True
16	<keras.layers.convolutional.Conv2D object at 0x7f271a15eb38>	block5_conv2	True
17	<keras.layers.convolutional.Conv2D object at 0x7f26c8371d68>	block5_conv3	True
18	<keras.layers.pooling.MaxPooling2D object at 0x7f26c8314b00>	block5_pool	True
19	<keras.layers.core.Flatten object at 0x7f26c828bda0>	flatten_1	True




0
20
40
60
80
100
120
140
0
25
50
75
100
125


0
25
50
75
100
125
0
50
100


block1_conv1
block1_conv2
block1_pool
block2_conv1

Model Performance metrics:

Accuracy: 0.776
Precision: 0.7769
Recall: 0.776
F1 Score: 0.7758

Model Classification report:

	precision	recall	f1-score	support
cat	0.76	0.80	0.78	500
dog	0.79	0.75	0.77	500
avg / total	0.78	0.78	0.78	1000

Prediction Confusion Matrix:

	Predicted: cat	Predicted: dog
Actual: cat	402	98
Actual: dog	126	374

Model Performance metrics:

Accuracy: 0.844
Precision: 0.844
Recall: 0.844
F1 Score: 0.844

Model Classification report:

	precision	recall	f1-score	support
cat	0.84	0.84	0.84	500
dog	0.84	0.84	0.84	500
avg / total	0.84	0.84	0.84	1000

Prediction Confusion Matrix:

	Predicted: cat	Predicted: dog
Actual: cat	422	78
Actual: dog	78	422

Model Performance metrics:

Accuracy: 0.888
Precision: 0.8898
Recall: 0.888
F1 Score: 0.8879

Model Classification report:

	precision	recall	f1-score	support
cat	0.92	0.85	0.88	500
dog	0.86	0.92	0.89	500
avg / total	0.89	0.89	0.89	1000

Prediction Confusion Matrix:

	Predicted: cat	Predicted: dog
Actual: cat	427	73
Actual: dog	39	461

Model Performance metrics:

Accuracy: 0.898
Precision: 0.8981
Recall: 0.898
F1 Score: 0.898

Model Classification report:

	precision	recall	f1-score	support
cat	0.89	0.91	0.90	500
dog	0.90	0.89	0.90	500
avg / total	0.90	0.90	0.90	1000

Prediction Confusion Matrix:

	Predicted: cat	Predicted: dog
Actual: cat	453	47
Actual: dog	55	445

Model Performance metrics:

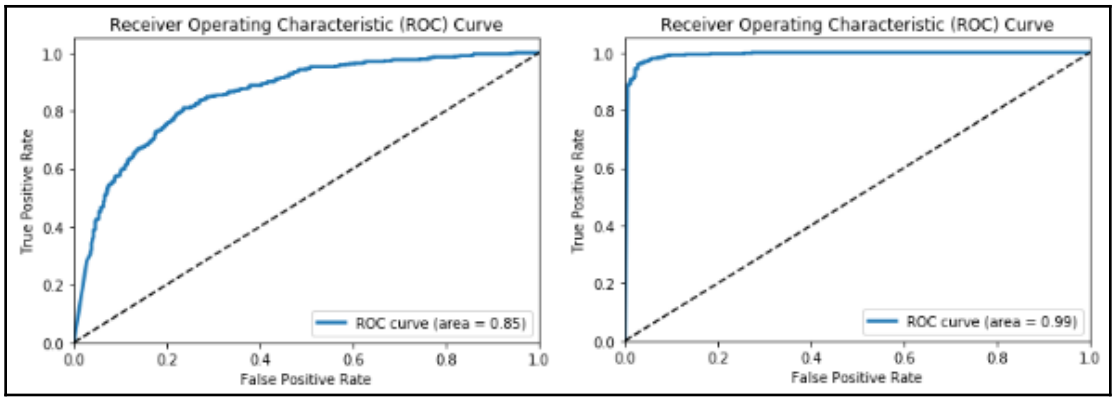
Accuracy: 0.961
Precision: 0.9611
Recall: 0.961
F1 Score: 0.961

Model Classification report:

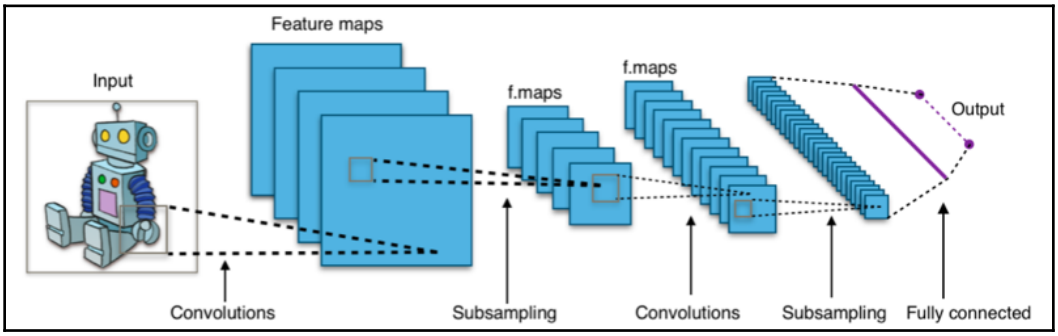
	precision	recall	f1-score	support
cat	0.97	0.95	0.96	500
dog	0.95	0.97	0.96	500
avg / total	0.96	0.96	0.96	1000

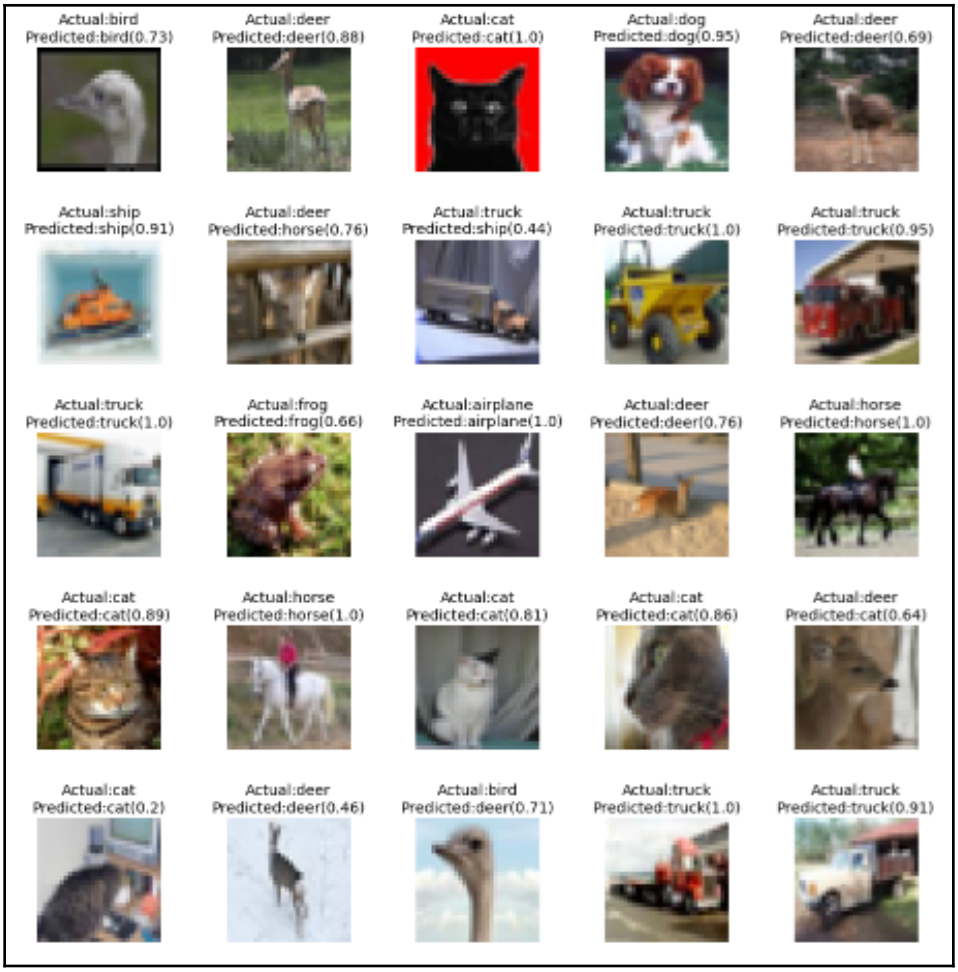
Prediction Confusion Matrix:

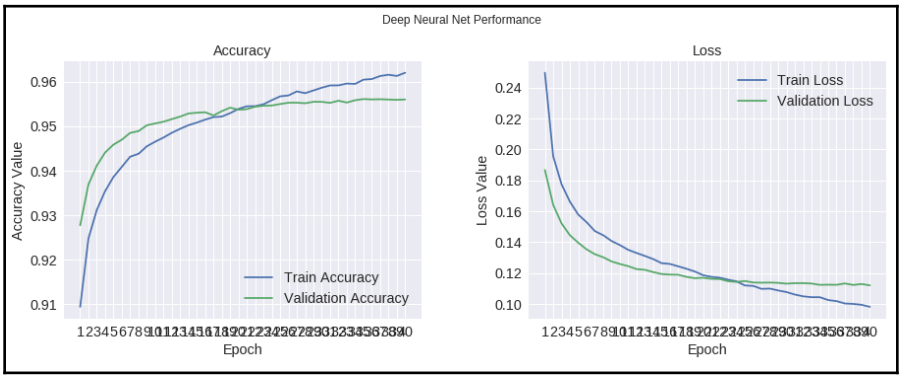
	Predicted: cat	Predicted: dog
Actual: cat	476	24
Actual: dog	15	485

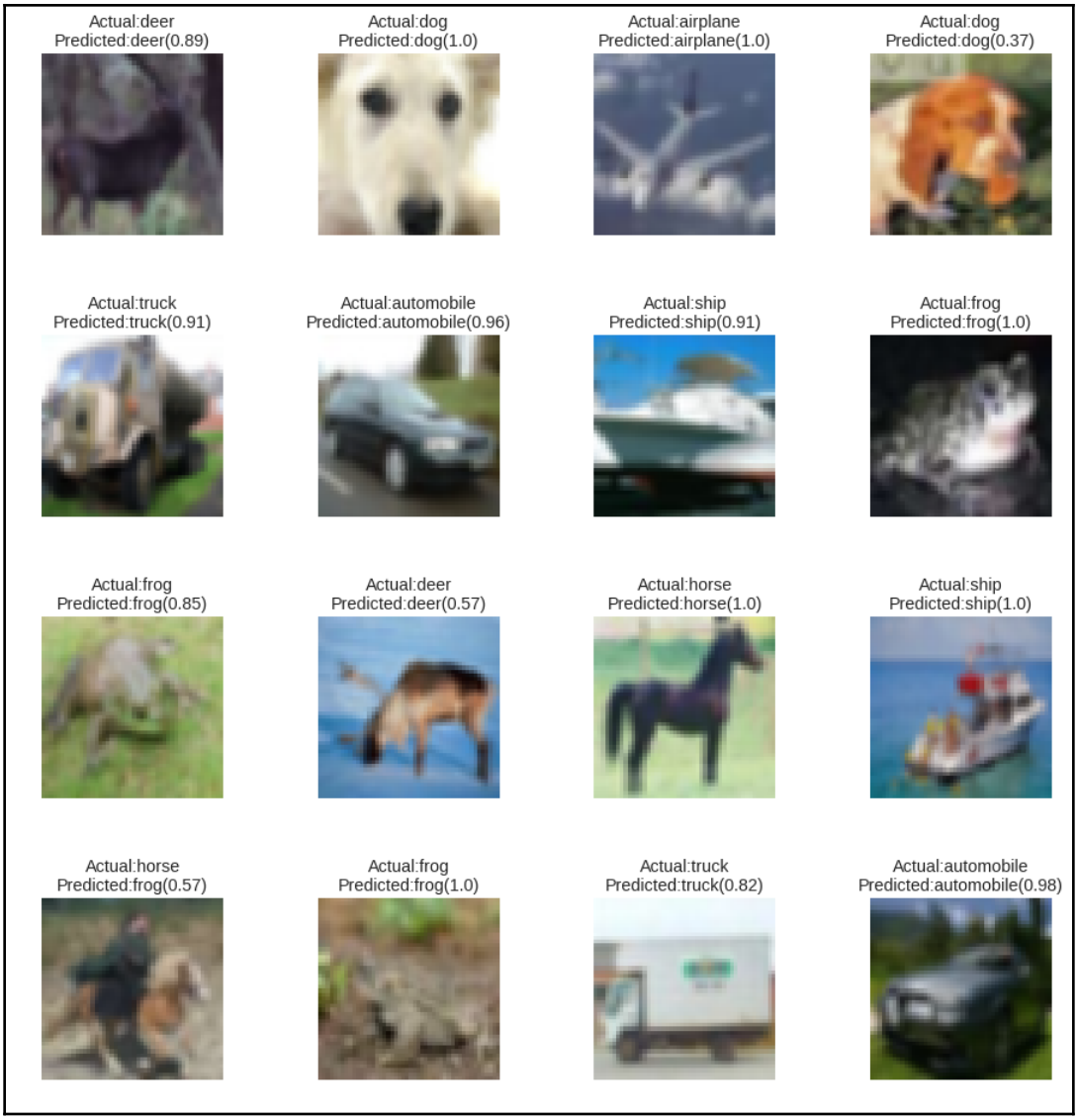

Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
ROC curve (area = 0.85)
Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
ROC curve (area = 0.99)

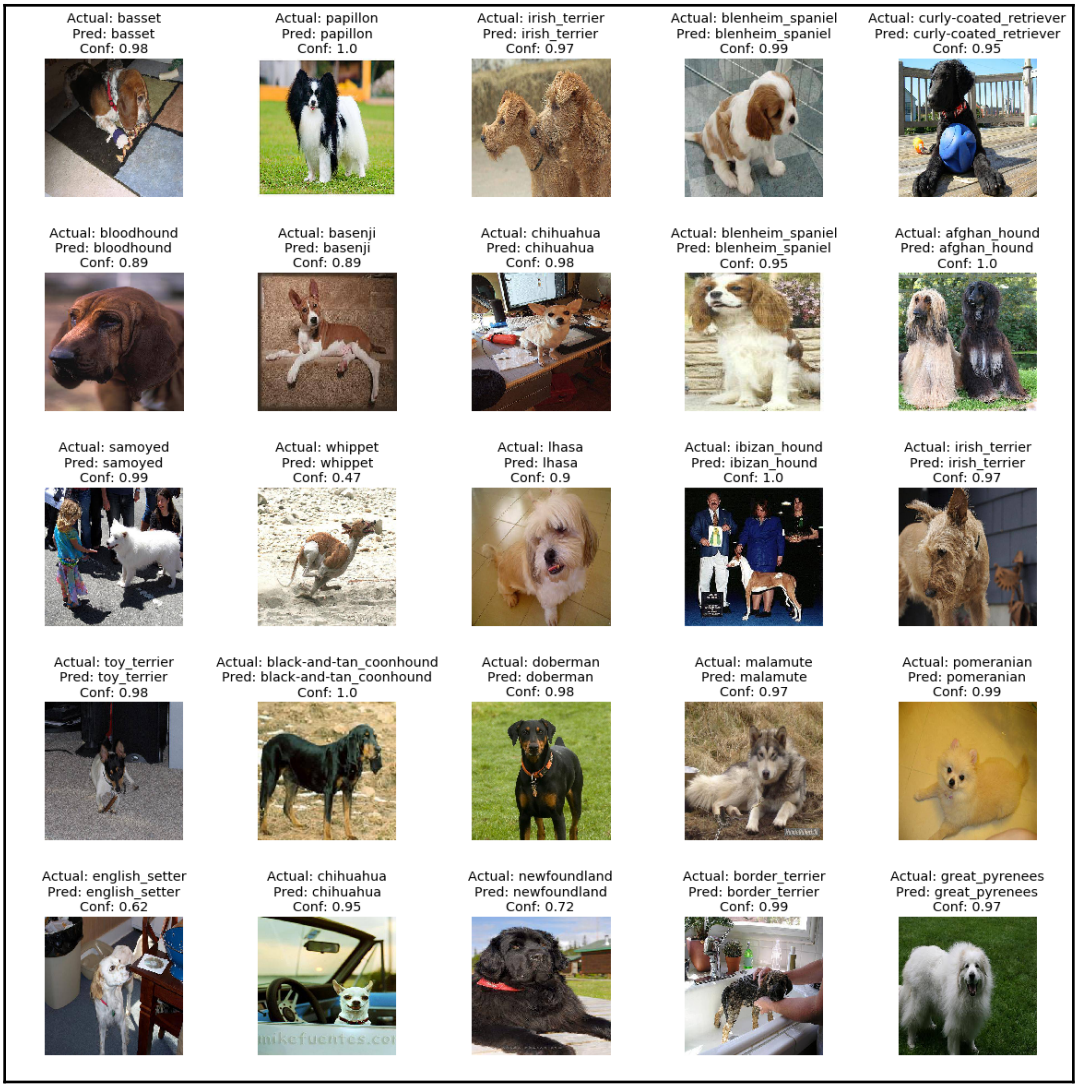
Chapter 6: Image Recognition and Classification




Actual:bird
Predicted:bird(0.73)
Actual:deer
Predicted:deer(0.88)
Actual:cat
Predicted:cat(1.0)
Actual:dog
Predicted:dog(0.95)
Actual:deer
Predicted:deer(0.69)
Actual:ship
Predicted:ship(0.91)
Actual:deer
Predicted:horse(0.76)
Actual:truck
Predicted:ship(0.44)
Actual:truck
Predicted:truck(1.0)
Actual:truck
Predicted:truck(0.95)
Actual:truck
Predicted:truck(1.0)
Actual:frog
Predicted:frog(0.66)
Actual:airplane
Predicted:airplane(1.0)
Actual:deer
Predicted:deer(0.76)
Actual:horse
Predicted:horse(1.0)
Actual:cat
Predicted:cat(0.89)
Actual:horse
Predicted:horse(1.0)
Actual:cat
Predicted:cat(0.81)
Actual:cat
Predicted:cat(0.86)
Actual:deer
Predicted:cat(0.64)
Actual:cat
Predicted:cat(0.2)
Actual:deer
Predicted:deer(0.46)
Actual:bird
Predicted:deer(0.71)
Actual:truck
Predicted:truck(1.0)
Actual:truck
Predicted:truck(0.91)


Deep Neural Net Performance
Accuracy
Accuracy Value
0.96
0.95
0.94
0.93
0.92
0.91
Train Accuracy
Validation Accuracy
Epoch
Loss
Loss Value
0.24
0.22
0.20
0.18
0.16
0.14
0.12
0.10
Train Loss
Validation Loss
Epoch


Actual:deer
Predicted:deer(0.89)
Actual:dog
Predicted:dog(1.0)
Actual:airplane
Predicted:airplane(1.0)
Actual:dog
Predicted:dog(0.37)
Actual:truck
Predicted:truck(0.91)
Actual:automobile
Predicted:automobile(0.96)
Actual:ship
Predicted:ship(0.91)
Actual:frog
Predicted:frog(1.0)
Actual:frog
Predicted:frog(0.85)
Actual:deer
Predicted:deer(0.57)
Actual:horse
Predicted:horse(1.0)
Actual:ship
Predicted:ship(1.0)
Actual:horse
Predicted:frog(0.57)
Actual:frog
Predicted:frog(1.0)
Actual:truck
Predicted:truck(0.82)
Actual:automobile
Predicted:automobile(0.98)


Actual: basset
Pred: basset
Conf: 0.98
Actual: papillon
Pred: papillon
Conf: 1.0
Actual: irish_terrier
Pred: irish_terrier
Conf: 0.97
Actual: blenheim_spaniel
Pred: blenheim_spaniel
Conf: 0.99
Actual: curly-coated_retriever
Pred: curly-coated_retriever
Conf: 0.95
Actual: bloodhound
Pred: bloodhound
Conf: 0.89
Actual: basenji
Pred: basenji
Conf: 0.89
Actual: chihuahua
Pred: chihuahua
Conf: 0.98
Actual: blenheim_spaniel
Pred: blenheim_spaniel
Conf: 0.95
Actual: afghan_hound
Pred: afghan_hound
Conf: 1.0
Actual: samoyed
Pred: samoyed
Conf: 0.99
Actual: whippet
Pred: whippet
Conf: 0.47
Actual: lhasa
Pred: lhasa
Conf: 0.9
Actual: ibizan_hound
Pred: ibizan_hound
Conf: 1.0
Actual: irish_terrier
Pred: irish_terrier
Conf: 0.97
Actual: toy_terrier
Pred: toy_terrier
Conf: 0.98
Actual: black-and-tan_coonhound
Pred: black-and-tan_coonhound
Conf: 1.0
Actual: doberman
Pred: doberman
Conf: 0.98
Actual: malamute
Pred: malamute
Conf: 0.97
Actual: pomeranian
Pred: pomeranian
Conf: 0.99
Actual: english_setter
Pred: english_setter
Conf: 0.62
Actual: chihuahua
Pred: chihuahua
Conf: 0.95
Actual: newfoundland
Pred: newfoundland
Conf: 0.72
Actual: border_terrier
Pred: border_terrier
Conf: 0.99
Actual: great_pyrenees
Pred: great_pyrenees
Conf: 0.97

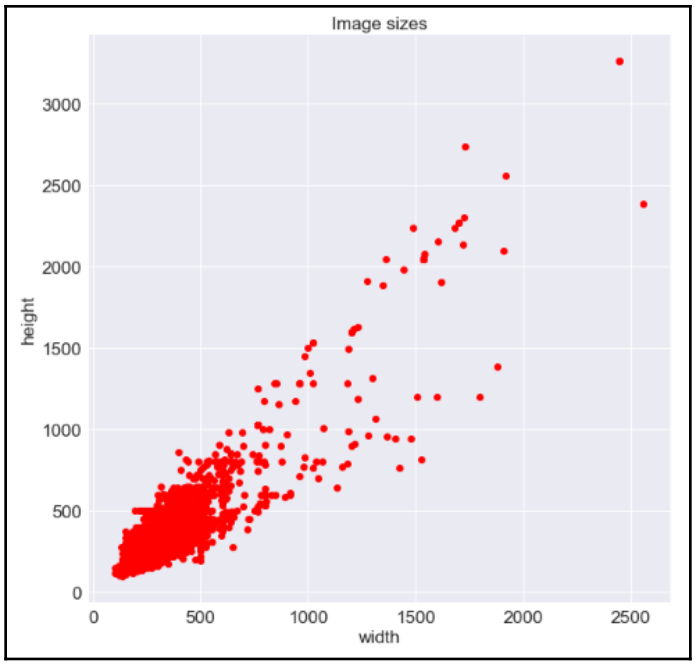
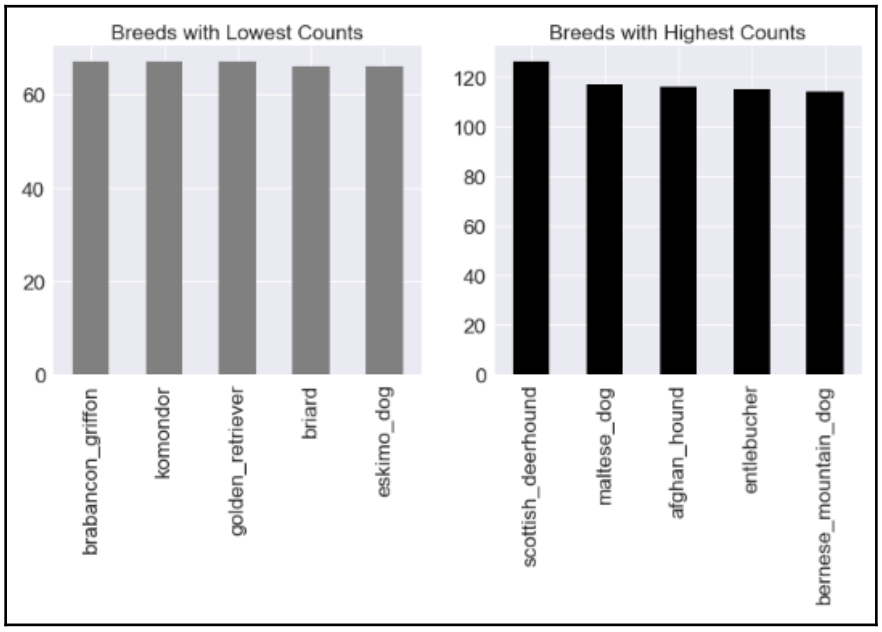
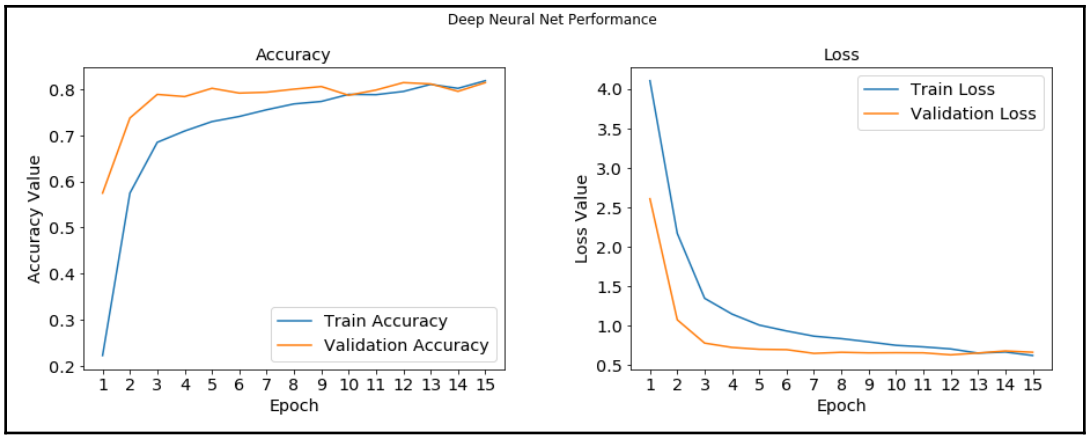
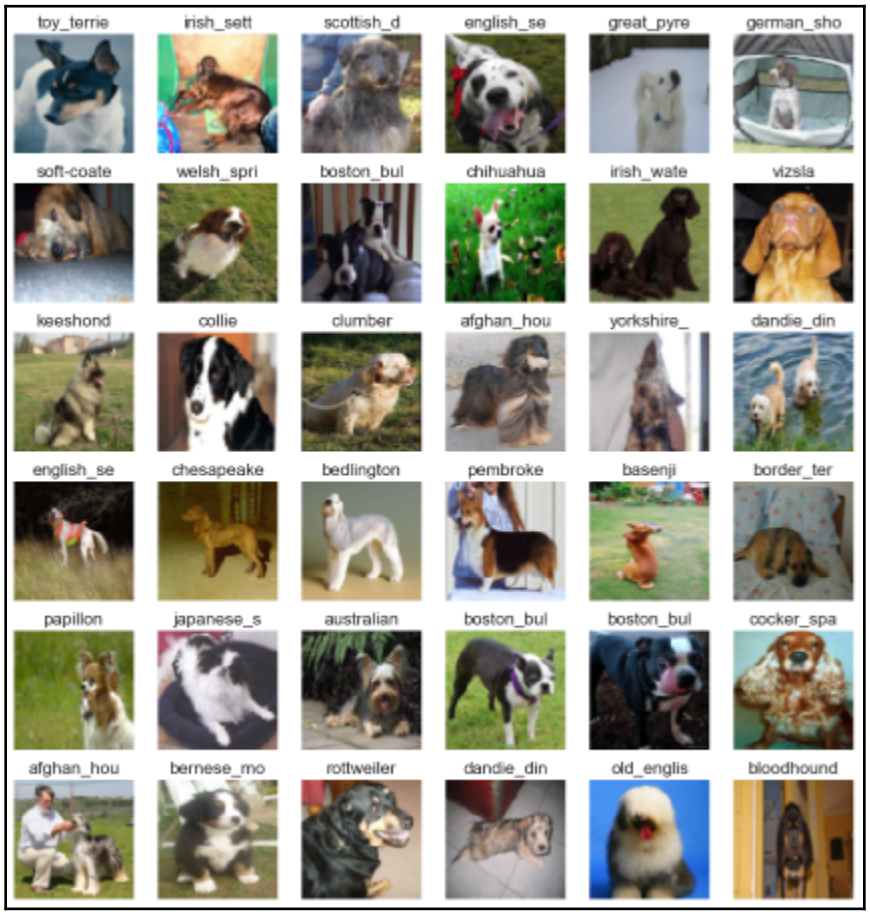


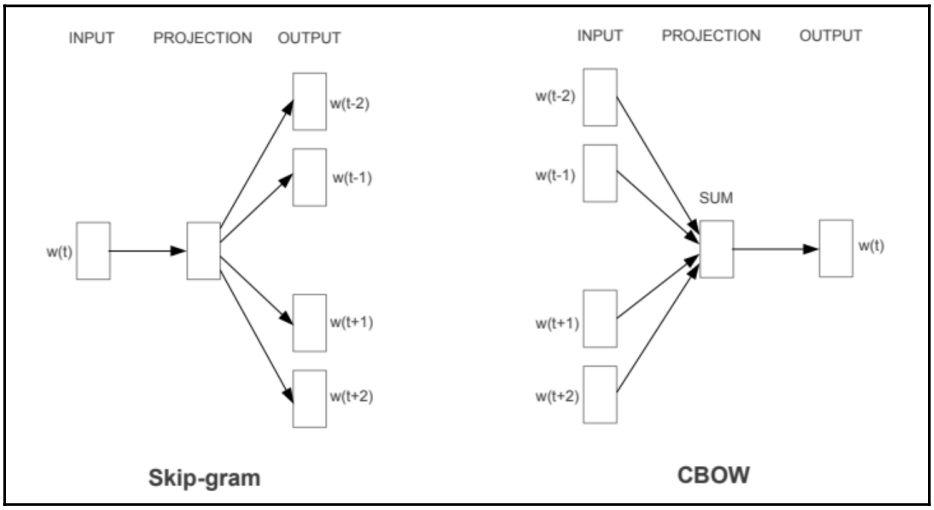
Image sizes
height
width
0
500
1000
1500
2000
2500
3000


Breeds with Lowest Counts
Breeds with Highest Counts
0
20
40
60
80
100
120
brabancon_griffon
komondor
golden_retriever
briard
eskimo_dog
scottish_deerhound
maltese_dog
afghan_hound
entlebucher
bernese_mountain_dog


Deep Neural Net Performance
Accuracy
Accuracy Value
Epoch
Train Accuracy
Validation Accuracy
Loss
Loss Value
Epoch
Train Loss
Validation Loss

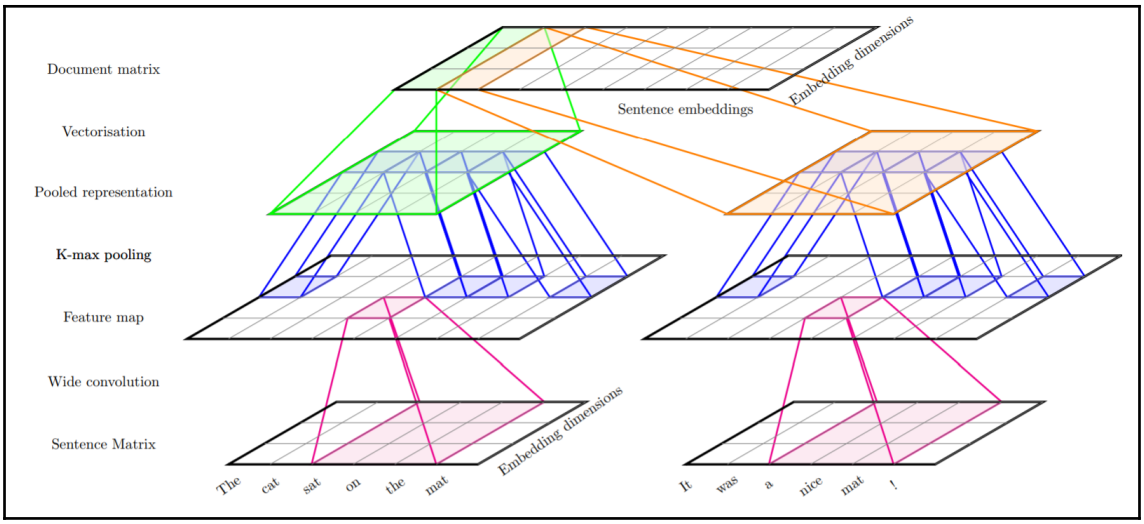

toy_terrie
irish_sett
scottish_d
english_se
great_pyre
german_sho
soft-coate
welsh_spri
boston_bul
chihuahua
irish_wate
vizsla
keeshond
collie
clumber
afghan_hou
yorkshire_
dandie_din
english_se
chesapeake
bedlington
pembroke
basenji
border_ter
papillon
japanese_s
australian
boston_bul
boston_bul
cocker_spa
afghan_hou
bernese_mo
rottweiler
dandie_din
old_englis
bloodhound

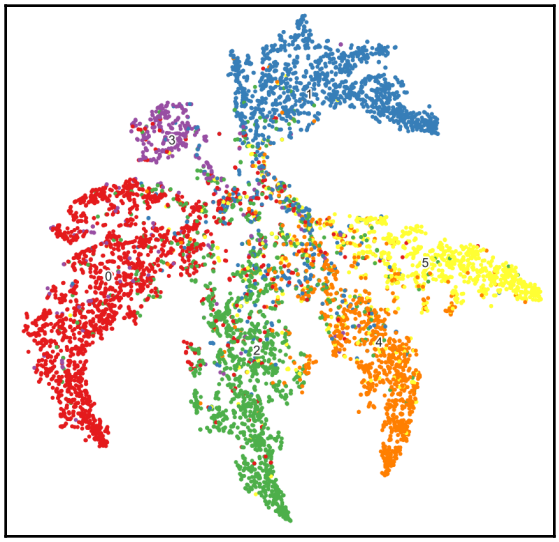
Chapter 7: Text Document Categorization



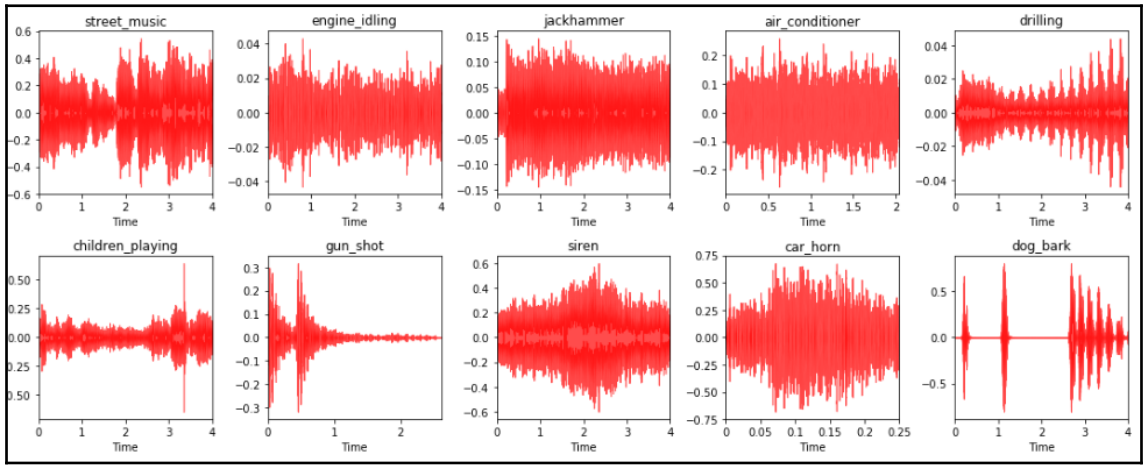
Out[12]:

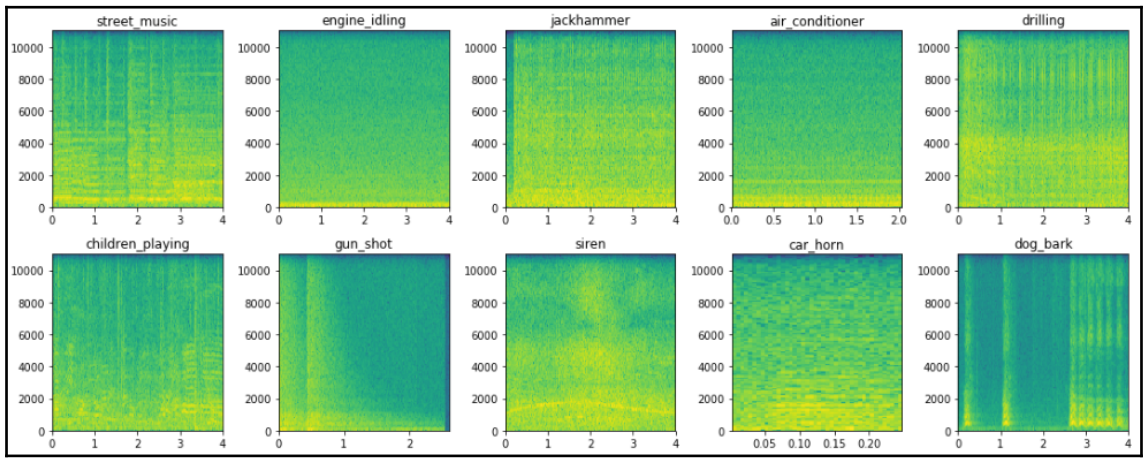
	0	1	2	3	4
bland	dull	lifeless	forgettable	uninspired	unconvincing
boring	dull	pointless	tedious	predictable	uninteresting
confused	irritated	puzzled	disturbed	frustrated	annoyed
disappointing	unsatisfying	disappointed	enjoyable	surprising	satisfying
fascinating	compelling	enthralling	captivating	unique	vivid
good	decent	great	nice	bad	fine
hilarious	funny	hysterical	priceless	comical	humorous
imaginative	inventive	innovative	ingenious	intricate	creative
nasty	sadistic	sleazy	gory	icky	vicious
predictable	clichéd	formulaic	contrived	implausible	dull
romantic	romance	screwball	bittersweet	sentimental	delightful
senseless	pointless	meaningless	disgusting	sickening	boring
sensitive	sincere	passionate	mature	delicate	confident
superior	inferior	weaker	truer	classier	maligned
unfortunate	unacceptable	disastrous	dubious	inadequate	important
violent	brutal	graphic	gruesome	sadistic	violence

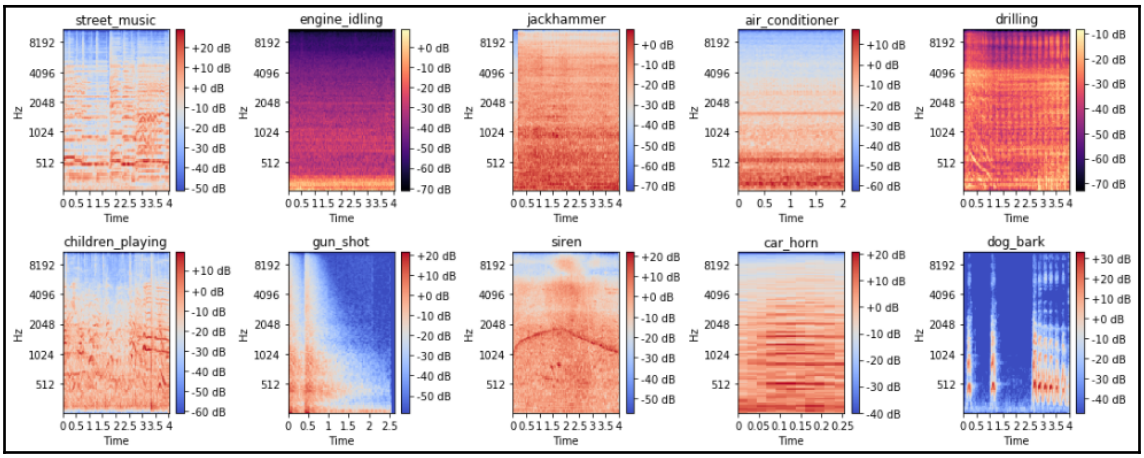

Document matrix
Vectorisation
Pooled representation
K-max pooling
Feature map
Wide convolution
Sentence Matrix
Sentence embeddings
Embedding dimensions
Embedding dimensions
The
cat
sat
on
the
mat
It
was
a
nice
mat
!

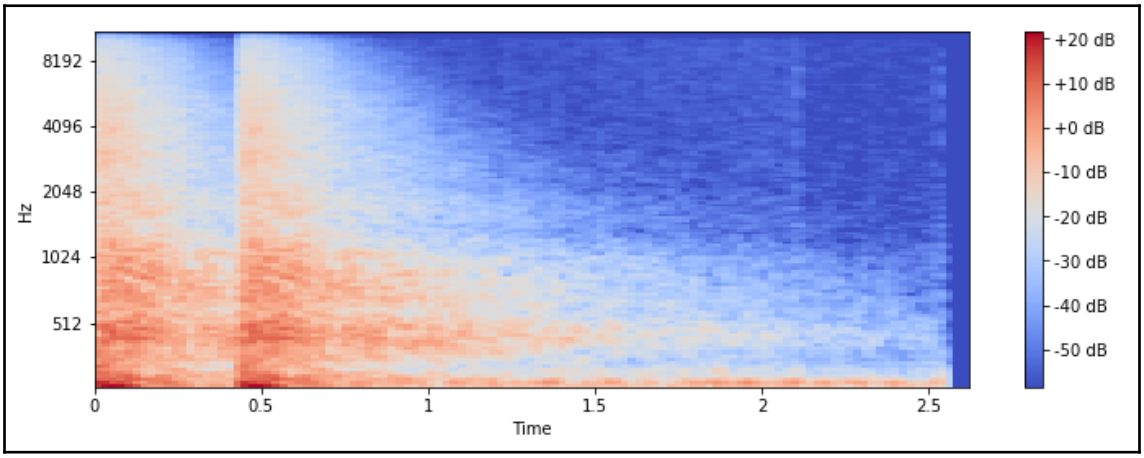

0
1
2
3
4
5

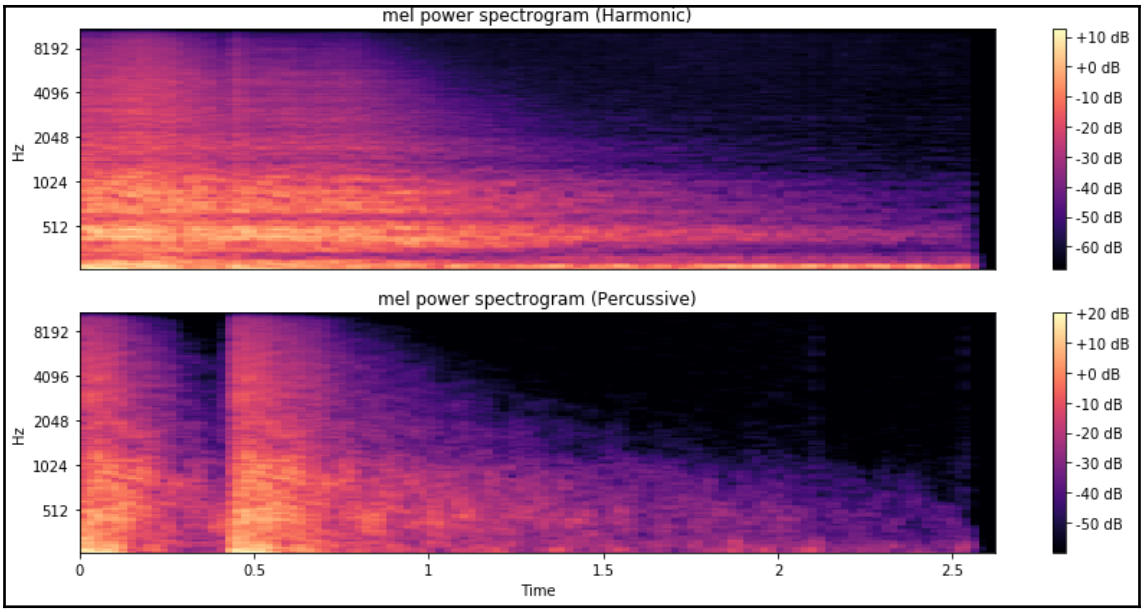
Chapter 8: Audio Event Identification and Classification

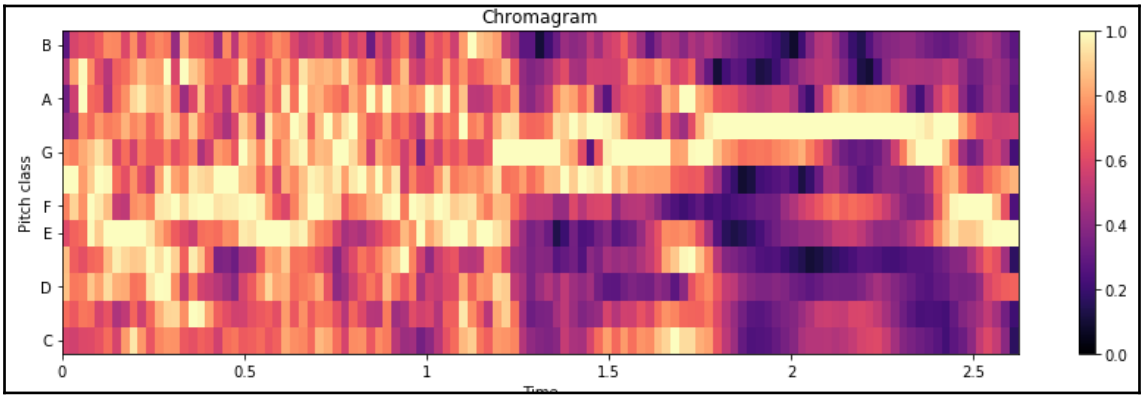


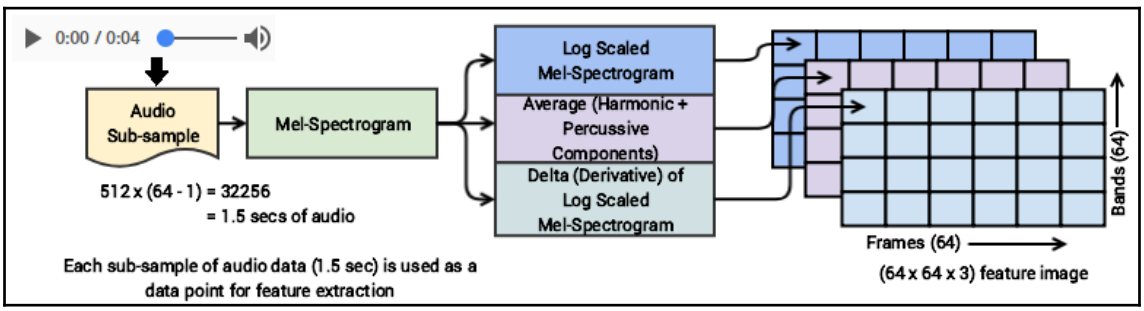


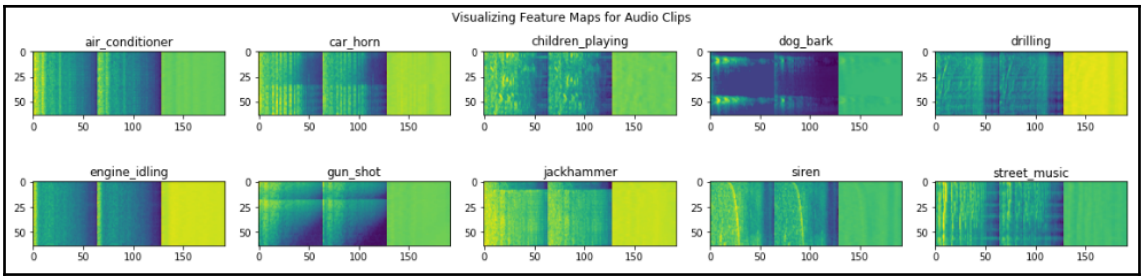

street_music
engine_idling
jackhammer
air_conditioner
drilling
children_playing
gun_shot
siren
car_horn
dog_bark
Hz
Time

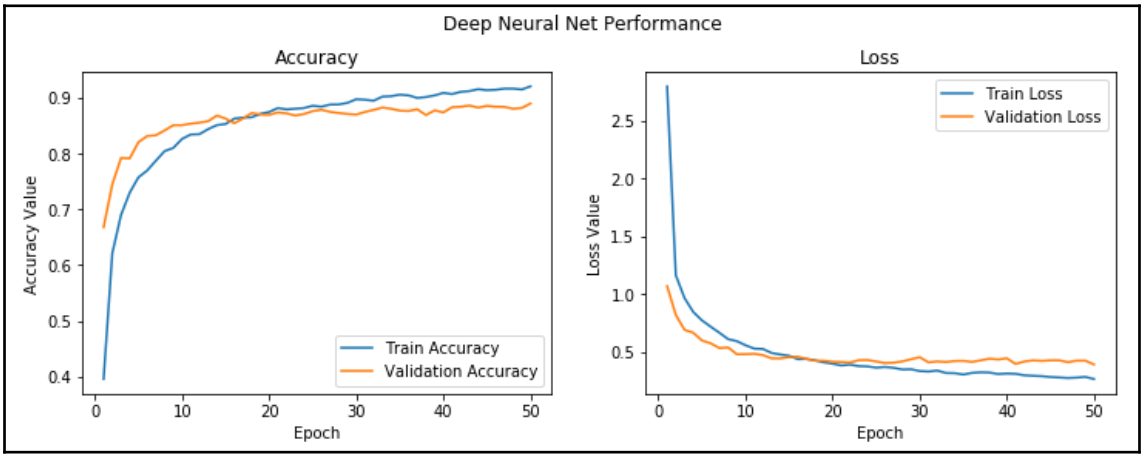

Hz
Time


mel power spectrogram (Harmonic)
8192
4096
2048
1024
512
Hz
+10 dB
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
mel power spectrogram (Percussive)
8192
4096
2048
1024
512
Hz
+20 dB
+10 dB
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
0
0.5
1
1.5
2
2.5
Time

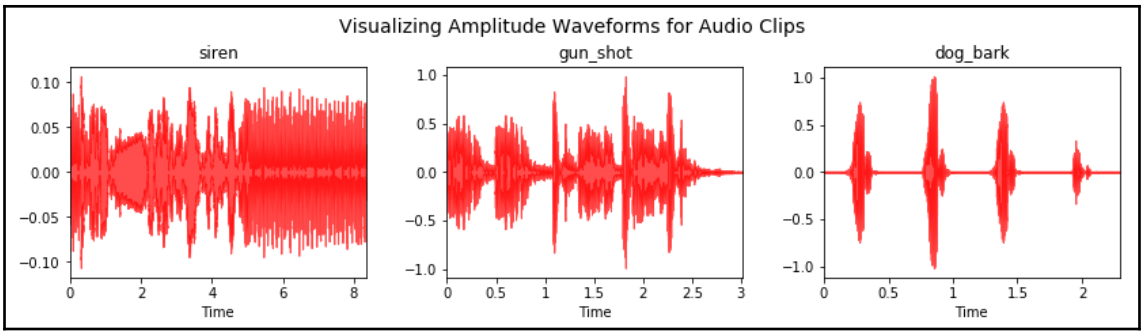

Chromagram
B
A
G
F
E
D
C
Pitch class
1.0
0.8
0.6
0.4
0.2
0.0
0
0.5
1
1.5
2
2.5

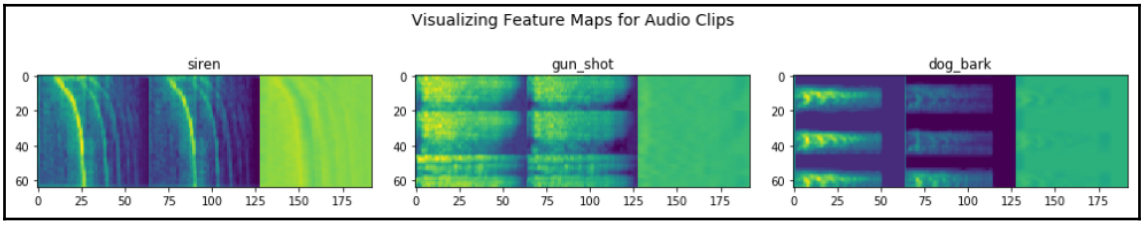

0:00 / 0:04
Audio Sub-sample
Mel-Spectrogram
Log Scaled Mel-Spectrogram
Average (Harmonic + Percussive Components)
Delta (Derivative) of Log Scaled Mel-Spectrogram
512 x (64 - 1) = 32256
= 1.5 secs of audio
Each sub-sample of audio data (1.5 sec) is used as a data point for feature extraction
Bands (64)
Frames (64)
(64 x 64 x 3) feature image





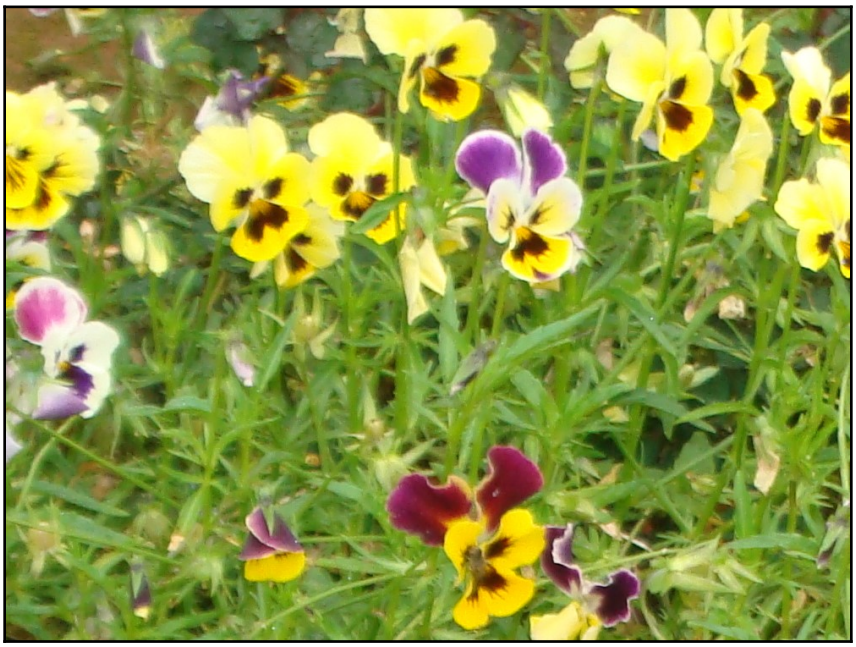
		Predicted:									
		car_horn	siren	drilling	gun_shot	children_playing	air_conditioner	jackhammer	engine_idling	dog_bark	street_music
Actual:	car_horn	137	4	15	0	3	3	4	1	2	19
	siren	1	705	5	0	7	18	1	2	6	5
	drilling	1	2	650	0	1	4	29	2	4	4
	gun_shot	0	0	1	67	0	0	0	1	2	0
	children_playing	2	11	13	1	592	14	2	7	31	77
	air_conditioner	2	0	6	1	8	768	7	10	2	9
	jackhammer	0	0	28	0	0	14	680	11	0	2
	engine_idling	1	0	3	0	4	10	4	707	8	8
	dog_bark	1	9	8	2	37	10	0	1	448	27
	street_music	12	15	12	0	64	18	10	7	14	656



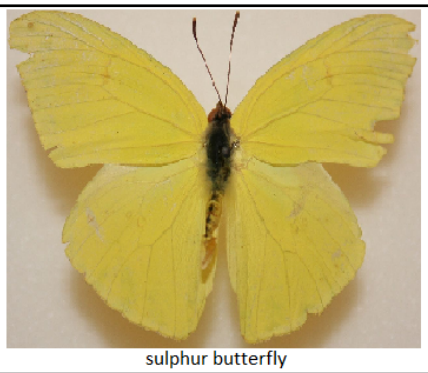


	Actual Sound	Location	Predicted Sound
0	siren	UrbanSound8K/test/sirenpolice.wav	siren
1	gun_shot	UrbanSound8K/test/gunfight.wav	gun_shot
2	dog_bark	UrbanSound8K/test/dog_bark.wav	dog_bark

Chapter 9: DeepDream

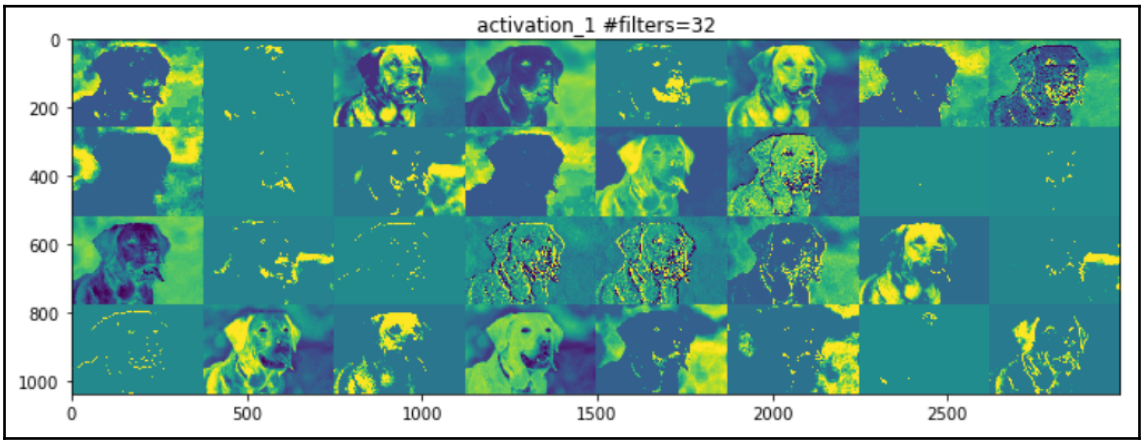


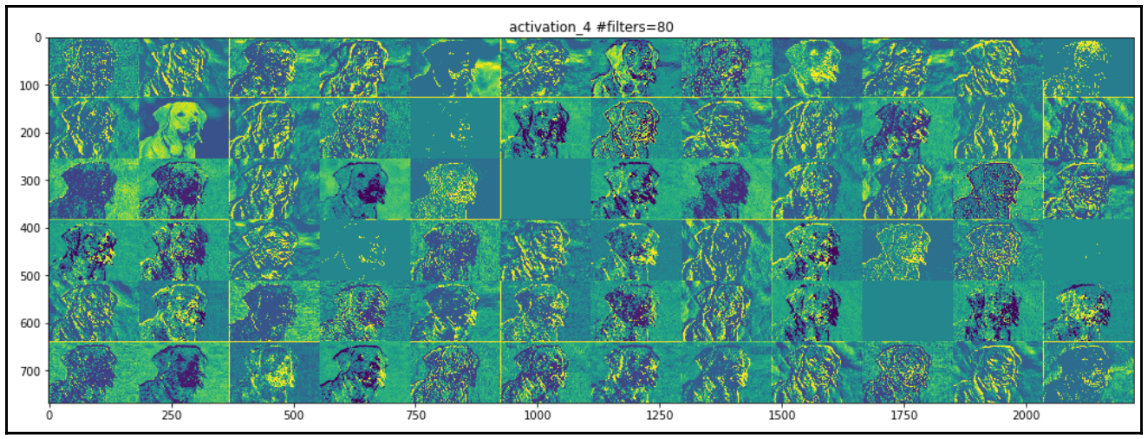
cabbage butterfly

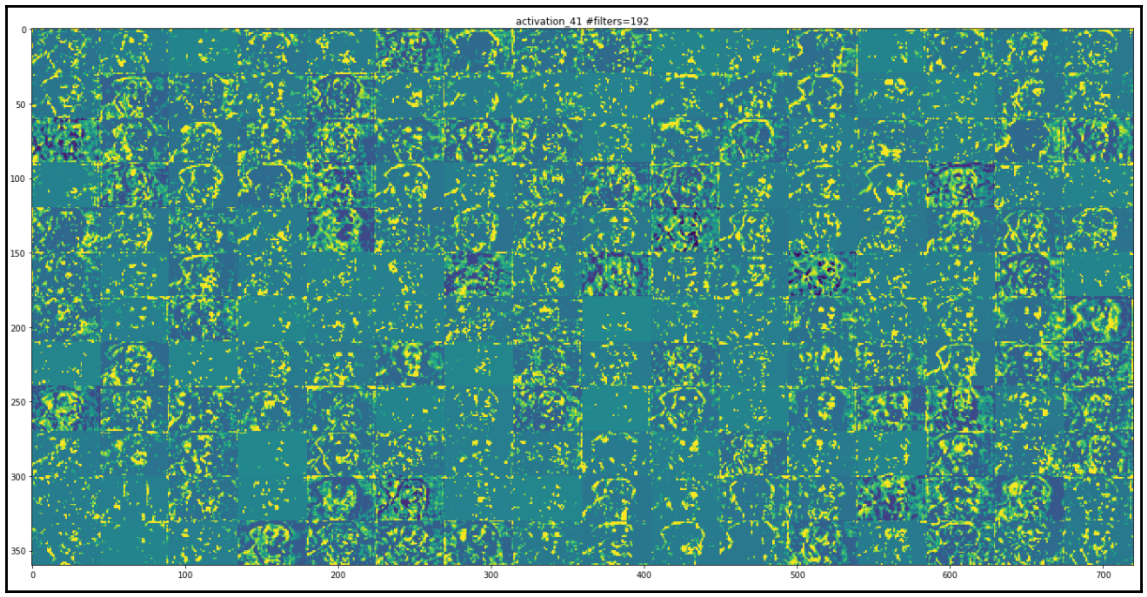


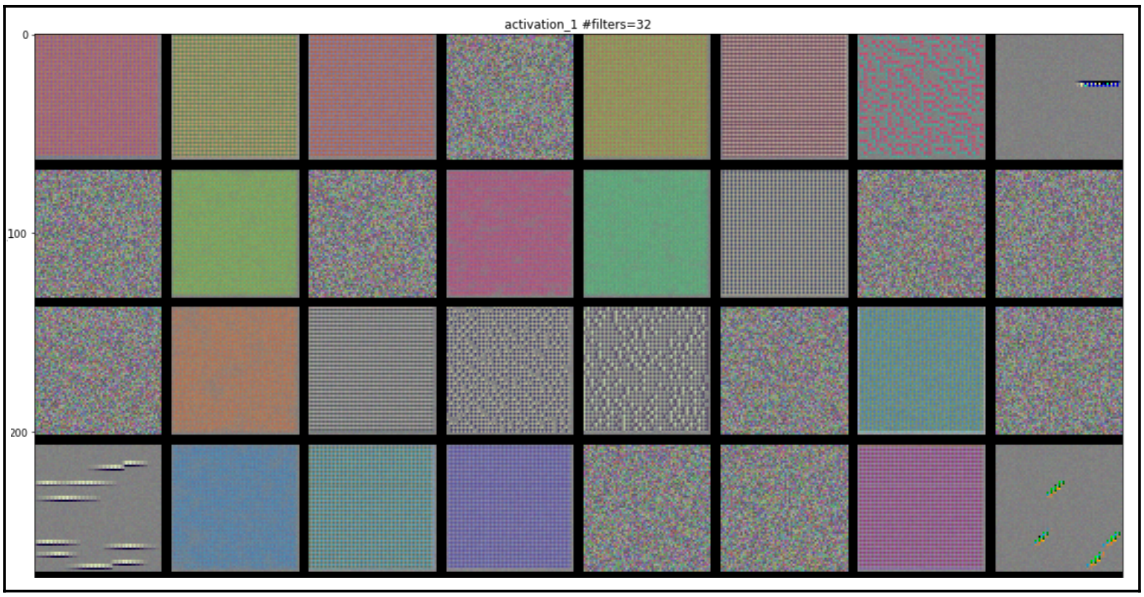
sulphur butterfly

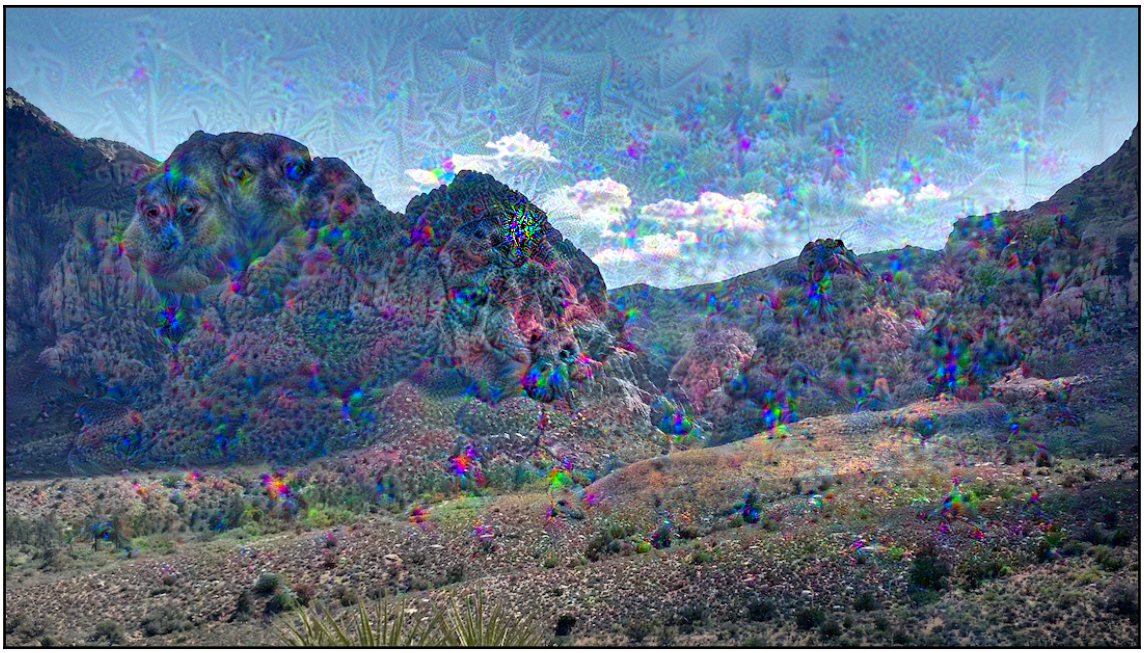
bee

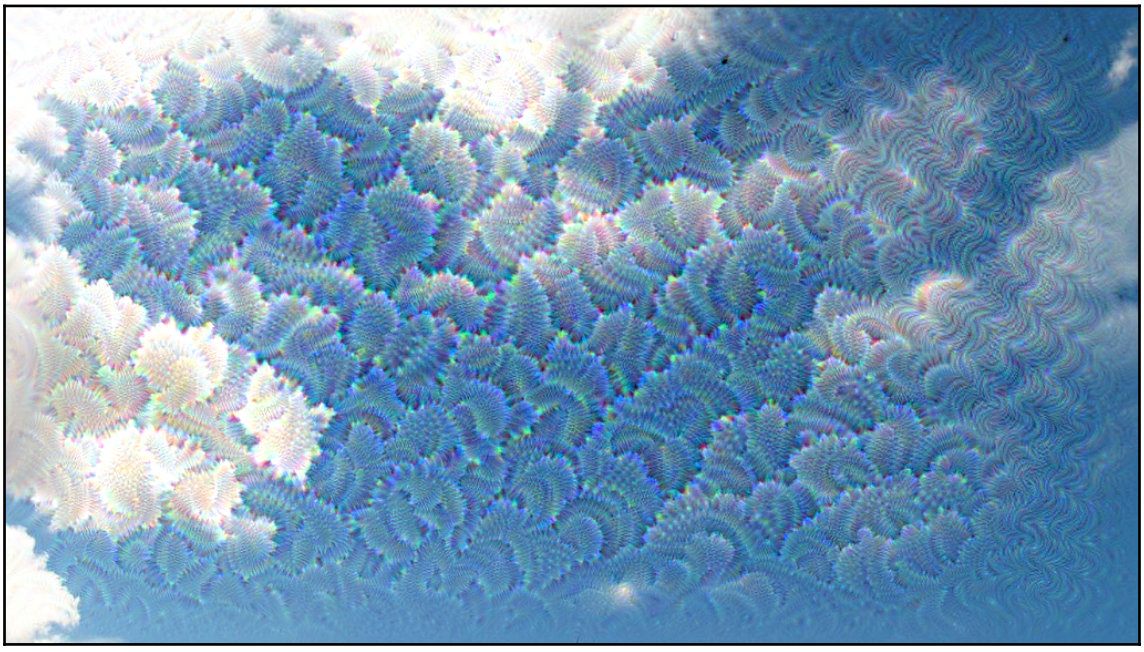

activation_1 #filters=32
0
200
400
600
800
1000
0
500
1000
1500
2000
2500


activation_4 #filters=80
0
100
200
300
400
500
600
700
0
250
500
750
1000
1250
1500
1750
2000

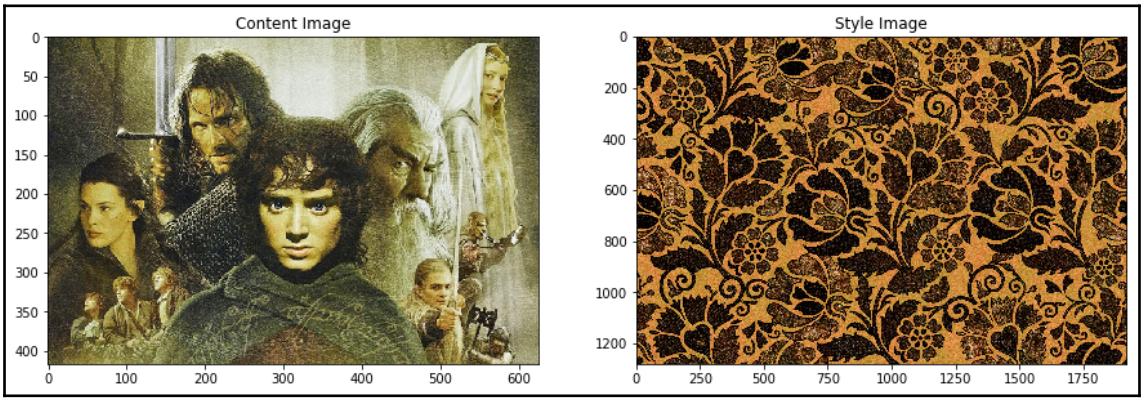

activation_41 #filters=192
0
50
100
150
200
250
300
350
0
100
200
300
400
500
600
700

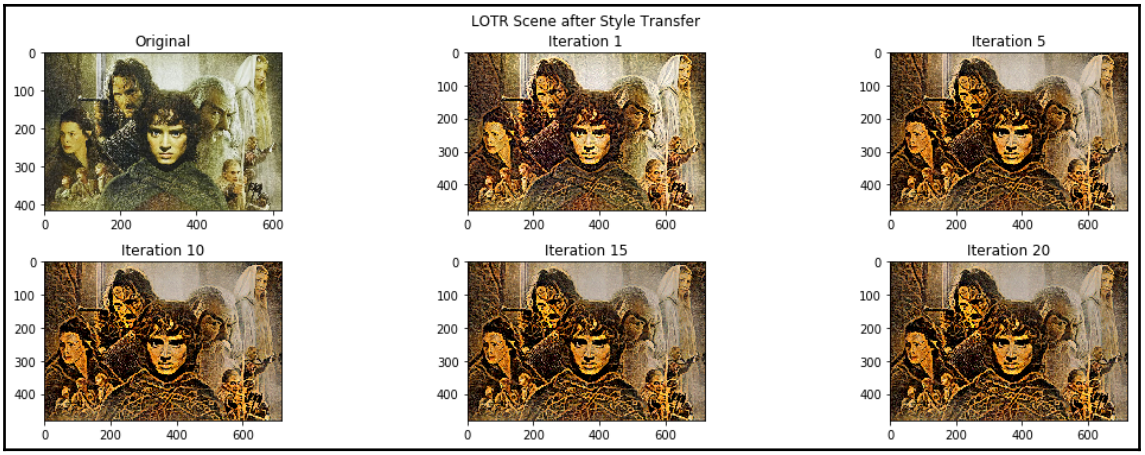

activation_1 #filters=32
0
100
200



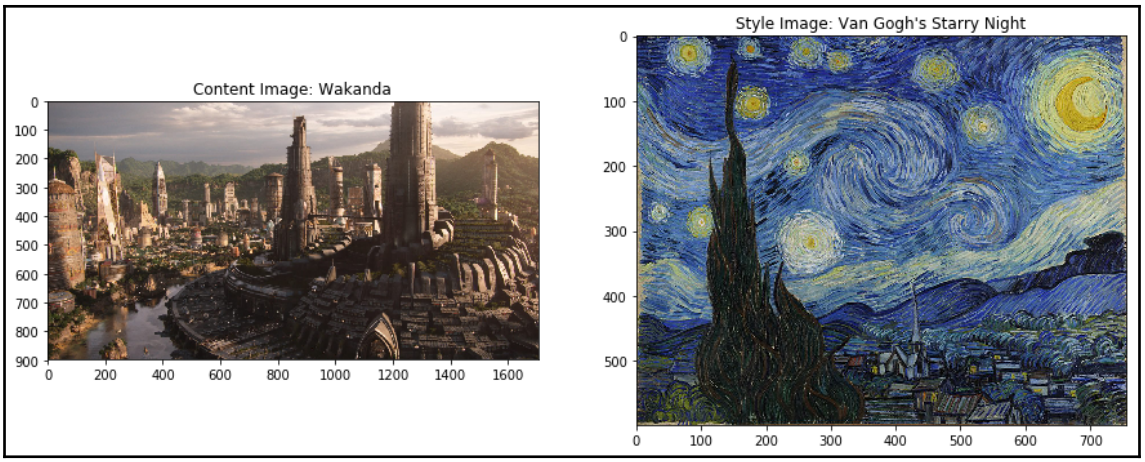


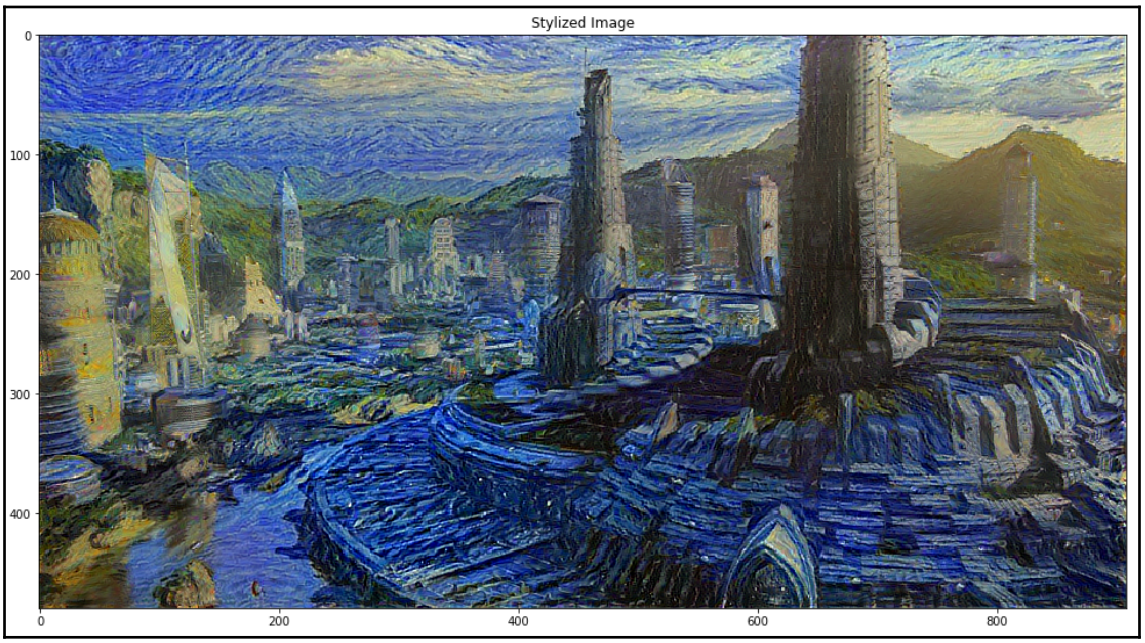
Chapter 10: Style Transfer




LOTR Scene after Style Transfer
Original
Iteration 1
Iteration 5
Iteration 10
Iteration 15
Iteration 20

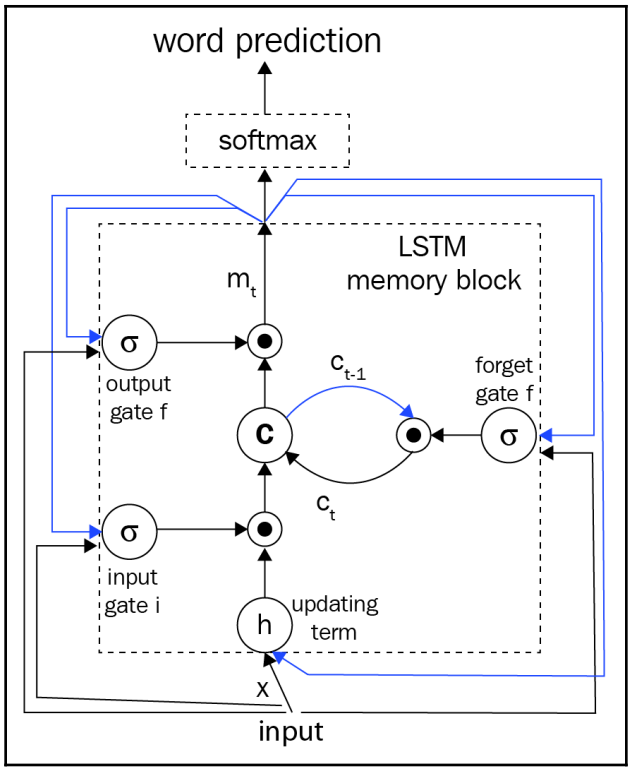

Stylized Image


Content Image: Wakanda
Style Image: Van Gogh's Starry Night

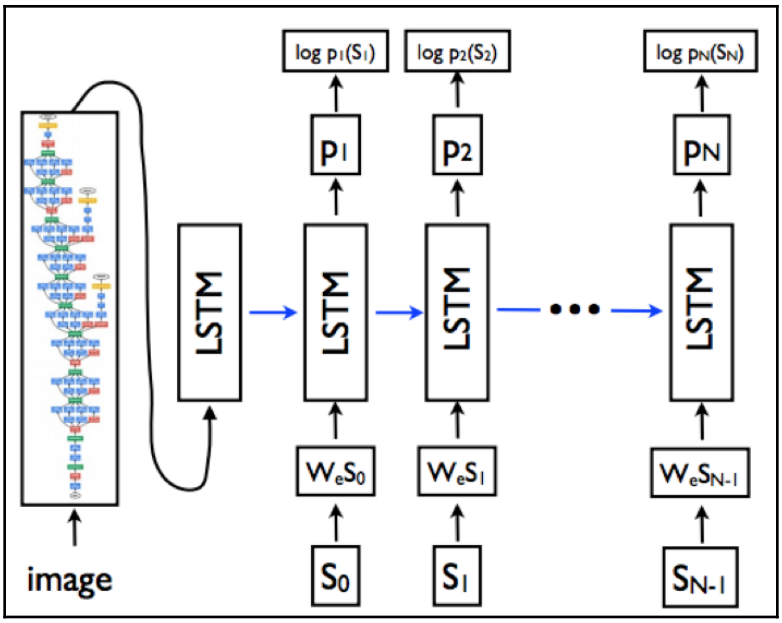

Stylized Image

Chapter 11: Automated Image Caption Generator

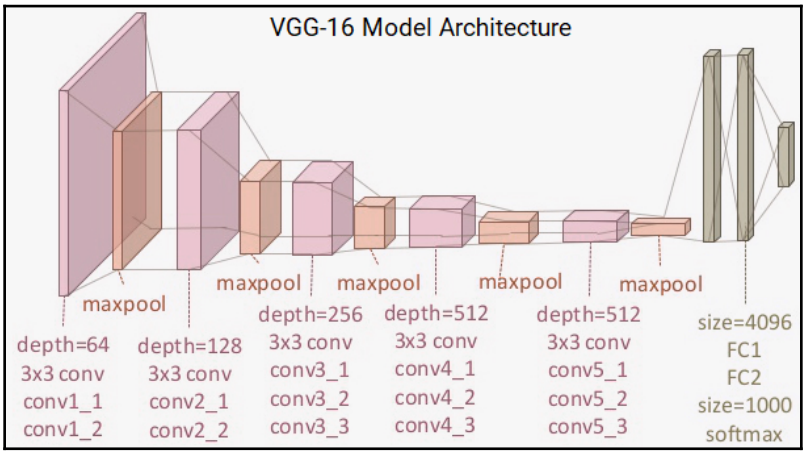
- A BMX bike rider in a black and red uniform on a dirt bike.
- A person on a bmx bike.
- A person wearing a black helmet rides a red bike through the woods.
- Biker with helmet riding red dirt bike in the woods.
- Dirt bike rider getting ready to start down the slope.



$$
\begin{aligned}
i_t &= \sigma(W_{ix}x_t + W_{im}m_{t-1}) \\
f_t &= \sigma(W_{fx}x_t + W_{fm}m_{t-1}) \\
o_t &= \sigma(W_{ox}x_t + W_{om}m_{t-1}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1}) \\
m_t &= o_t \odot c_t \\
p_{t+1} &= \text{Softmax}(m_t)
\end{aligned}
$$



$$
\begin{aligned}
x_{-1} &= \text{CNN}(I) \\
x_t &= W_e S_t, \quad t \in \{0 \dots N-1\} \\
p_{t+1} &= \text{LSTM}(x_t), \quad t \in \{0 \dots N-1\}
\end{aligned}
$$



	caption	image
0	A black dog is running after a white dog in th...	2513260012_03d33305cf.jpg
1	Black dog chasing brown dog through snow	2513260012_03d33305cf.jpg
2	Two dogs chase each other across the snowy gro...	2513260012_03d33305cf.jpg
3	Two dogs play together in the snow .	2513260012_03d33305cf.jpg
4	Two dogs running through a low lying body of w...	2513260012_03d33305cf.jpg
5	A little baby plays croquet .	2903617548_d3e38d7f88.jpg
6	A little girl plays croquet next to a truck .	2903617548_d3e38d7f88.jpg
7	The child is playing croquette by the truck .	2903617548_d3e38d7f88.jpg
8	The kid is in front of a car with a put and a ...	2903617548_d3e38d7f88.jpg
9	The little boy is playing with a croquet hamme...	2903617548_d3e38d7f88.jpg

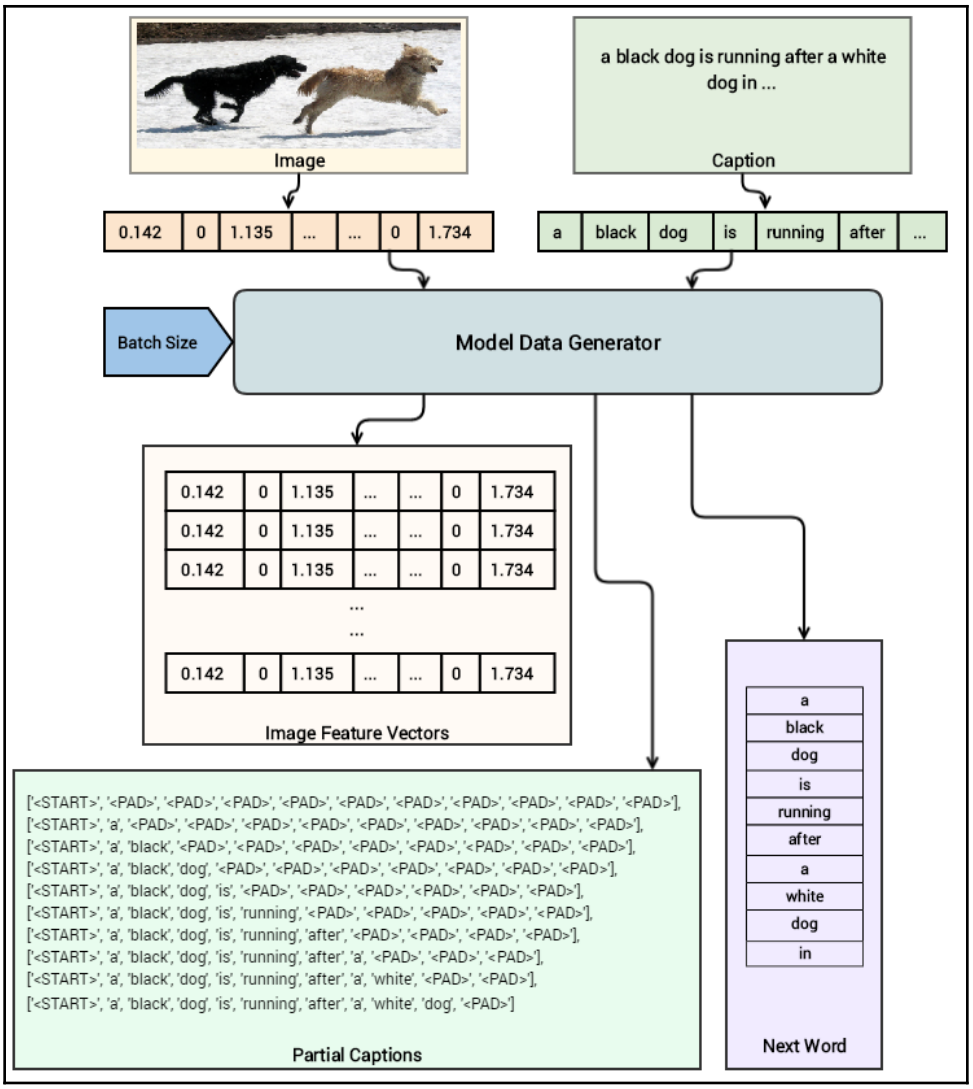
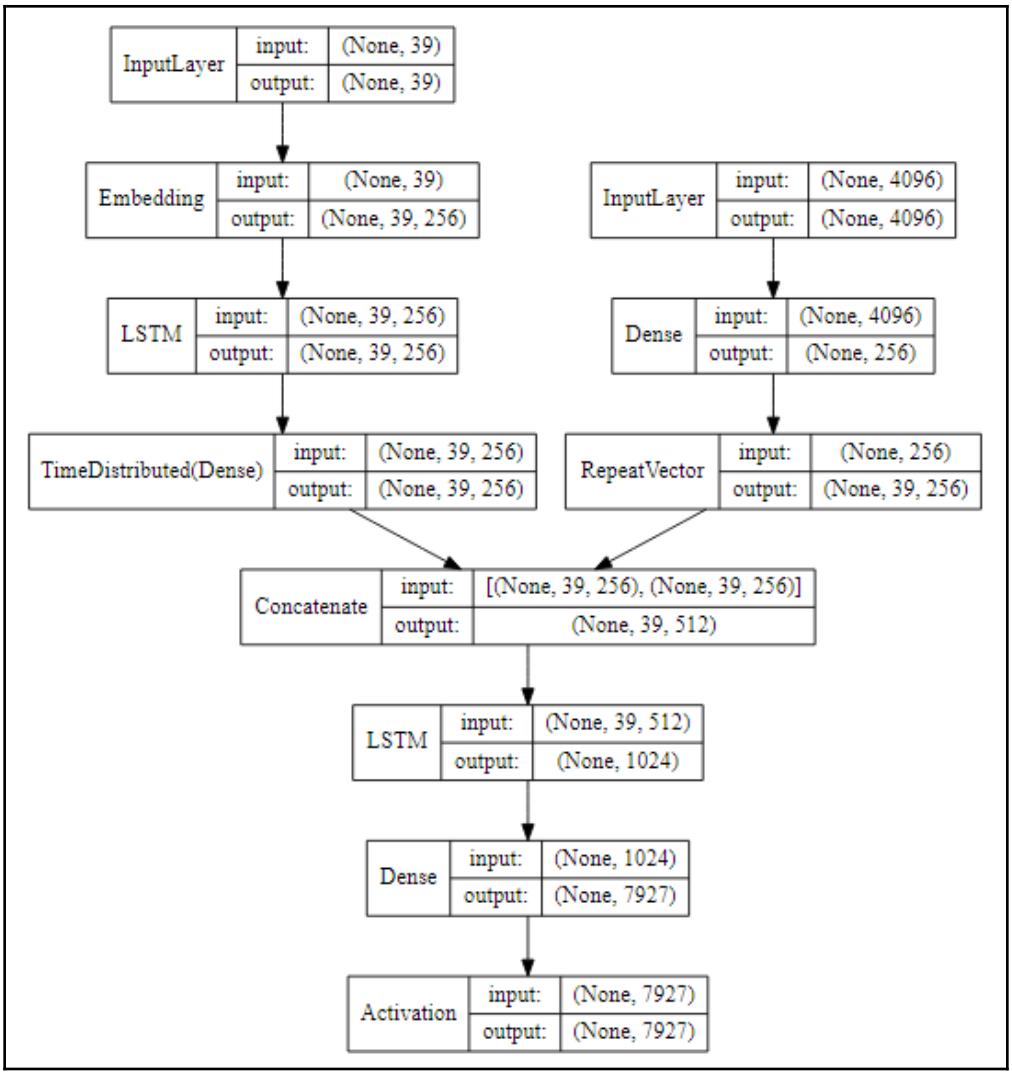
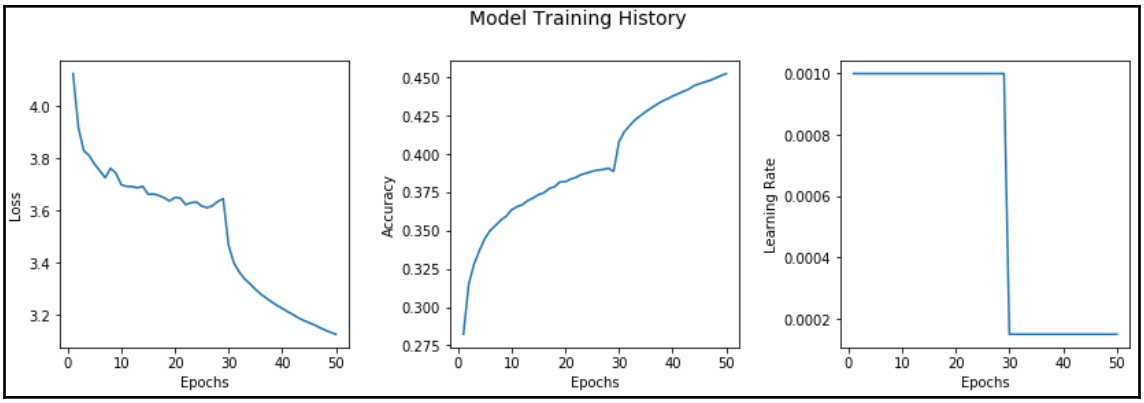


Image
a black dog is running after a white dog in ...
Caption
0.142 0 1.135 ... ... 0 1.734
a black dog is running after ...
Batch Size
Model Data Generator
0.142 0 1.135 ... ... 0 1.734
0.142 0 1.135 ... ... 0 1.734
0.142 0 1.135 ... ... 0 1.734
...
...
0.142 0 1.135 ... ... 0 1.734
Image Feature Vectors
['<START>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>'],
['<START>', 'a', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>'],
['<START>', 'a', 'black', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>'],
['<START>', 'a', 'black', 'dog', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>'],
['<START>', 'a', 'black', 'dog', 'is', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>'],
['<START>', 'a', 'black', 'dog', 'is', 'running', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>'],
['<START>', 'a', 'black', 'dog', 'is', 'running', 'after', '<PAD>', '<PAD>', '<PAD>', '<PAD>'],
['<START>', 'a', 'black', 'dog', 'is', 'running', 'after', 'a', '<PAD>', '<PAD>', '<PAD>'],
['<START>', 'a', 'black', 'dog', 'is', 'running', 'after', 'a', 'white', '<PAD>', '<PAD>'],
['<START>', 'a', 'black', 'dog', 'is', 'running', 'after', 'a', 'white', 'dog', '<PAD>']
Partial Captions
a
black
dog
is
running
after
a
white
dog
in
Next Word

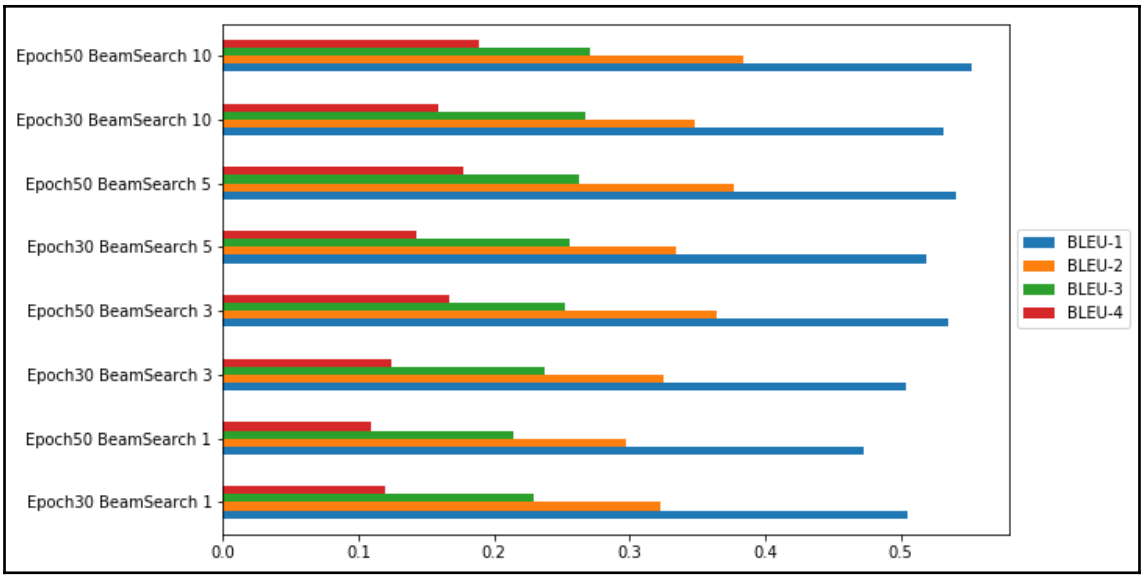
Layer (type)	Output Shape	Param #	Connected to
embedding_1_input (InputLayer)	(None, 39)	0	
dense_1_input (InputLayer)	(None, 4096)	0	
embedding_1 (Embedding)	(None, 39, 256)	2029312	embedding_1_input[0][0]
dense_1 (Dense)	(None, 256)	1048832	dense_1_input[0][0]
lstm_1 (LSTM)	(None, 39, 256)	525312	embedding_1[0][0]
repeat_vector_1 (RepeatVector)	(None, 39, 256)	0	dense_1[0][0]
time_distributed_1 (TimeDistribu	(None, 39, 256)	65792	lstm_1[0][0]
concatenate_1 (Concatenate)	(None, 39, 512)	0	repeat_vector_1[0][0] time_distributed_1[0][0]
lstm_2 (LSTM)	(None, 1024)	6295552	concatenate_1[0][0]
dense_3 (Dense)	(None, 7927)	8125175	lstm_2[0][0]
activation_1 (Activation)	(None, 7927)	0	dense_3[0][0]

Total params: 18,089,975
Trainable params: 18,089,975
Non-trainable params: 0


InputLayer
input: (None, 39)
output: (None, 39)
Embedding
input: (None, 39)
output: (None, 39, 256)
InputLayer
input: (None, 4096)
output: (None, 4096)
LSTM
input: (None, 39, 256)
output: (None, 39, 256)
Dense
input: (None, 4096)
output: (None, 256)
TimeDistributed(Dense)
input: (None, 39, 256)
output: (None, 39, 256)
RepeatVector
input: (None, 256)
output: (None, 39, 256)
Concatenate
input: [(None, 39, 256), (None, 39, 256)]
output: (None, 39, 512)
LSTM
input: (None, 39, 512)
output: (None, 1024)
Dense
input: (None, 1024)
output: (None, 7927)
Activation
input: (None, 7927)
output: (None, 7927)



	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Epoch30 BeamSearch 1	0.504957	0.322464	0.229623	0.120146
Epoch50 BeamSearch 1	0.472661	0.296896	0.213849	0.109014
Epoch30 BeamSearch 3	0.503568	0.325535	0.237255	0.124127
Epoch50 BeamSearch 3	0.535221	0.364593	0.252667	0.167056
Epoch30 BeamSearch 5	0.519058	0.334287	0.255741	0.143111
Epoch50 BeamSearch 5	0.541103	0.377383	0.262667	0.177056
Epoch30 BeamSearch 10	0.531228	0.348353	0.267133	0.158336
Epoch50 BeamSearch 10	0.551929	0.384170	0.270649	0.188850


Epoch50 BeamSearch 10
Epoch30 BeamSearch 10
Epoch50 BeamSearch 5
Epoch30 BeamSearch 5
Epoch50 BeamSearch 3
Epoch30 BeamSearch 3
Epoch50 BeamSearch 1
Epoch30 BeamSearch 1
0.0
0.1
0.2
0.3
0.4
0.5
BLEU-1
BLEU-2
BLEU-3
BLEU-4

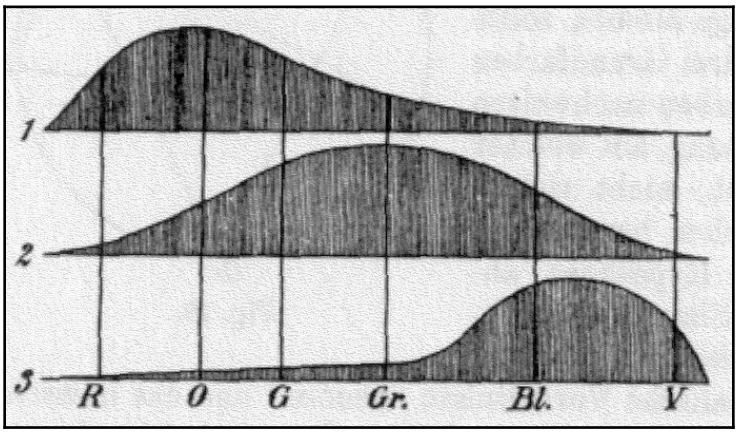

Automated Image Captioning: Outdoor Scenes 1
Caption(ep30): a little girl wearing a red shirt is swinging on a park swing
Caption(ep50): a little girl is swinging on a swing
Caption(ep30): a brown dog is running on the shore of the ocean
Caption(ep50): a brown dog is running in the sand at the beach
Caption(ep30): a black and white dog is playing with a red ball in the grass
Caption(ep50): a white dog is running through a field of green grass
Caption(ep30): a group of people stand on a road in the middle of a group of people
Caption(ep50): a group of people stand on a road looking at each other
Caption(ep30): a group of people are standing in front of a crowd of people
Caption(ep50): a group of people stand on a sidewalk near a crowd of people
Caption(ep30): a brown dog and a white dog are playing in the snow
Caption(ep50): two dogs play in the snow

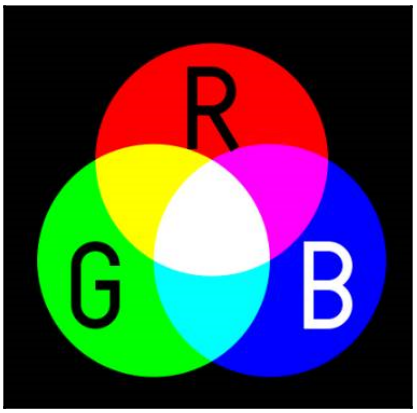

Automated Image Captioning: Outdoor Scenes 2
Caption(ep30): a man in a blue shirt is in the water with a boat in the background
Caption(ep50): a man in a blue shirt and a boat in the water
Caption(ep30): a man is riding a dirt bike in the woods
Caption(ep50): a man rides a dirt bike in the woods
Caption(ep30): a man is climbing a rock rock rock wall
Caption(ep50): a man in a red shirt is climbing a rock wall
Caption(ep30): a man in a blue jacket is playing in the snow
Caption(ep50): a man in a blue jacket is playing in the snow
Caption(ep30): a man in a red jacket is standing in the snow
Caption(ep50): a man in a blue jacket is skiing down a snowy mountain
Caption(ep30): a surfer in a blue wetsuit is surfing on a wave
Caption(ep50): a man in a blue wetsuit is riding a wave into the ocean

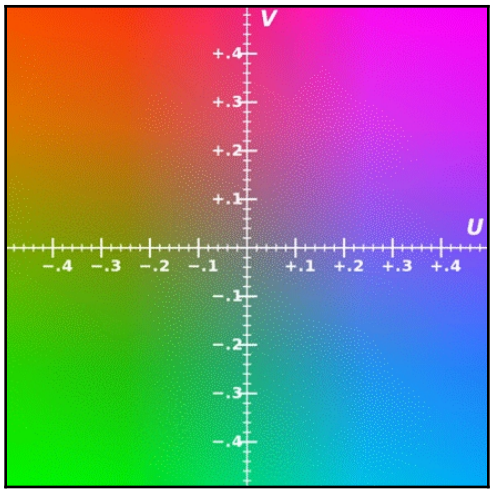

Automated Image Captioning: Sports Scenes 1
Caption(ep30): a baseball player is playing soccer
Caption(ep50): a baseball player in a white uniform is playing baseball
Caption(ep30): a man in a blue uniform is holding a soccer ball
Caption(ep50): two men playing basketball
Caption(ep30): a football player in a red jersey is playing football
Caption(ep50): a football player in red is being tackled by a player in a white jersey
Caption(ep30): the hockey player is racing down the ice
Caption(ep50): a hockey player looks at the goal
Caption(ep30): a race car is driving around a car
Caption(ep50): a race car is driving on a dirt track
Caption(ep30): a baseball player in a white uniform is playing tennis
Caption(ep50): a tennis player in a white uniform is playing tennis

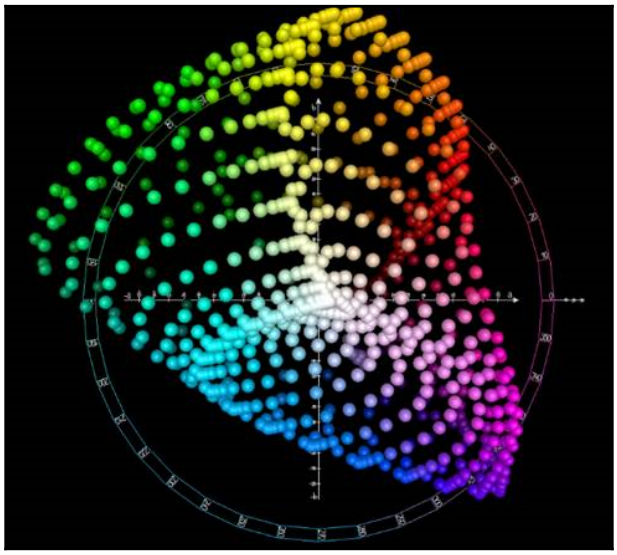

Automated Image Captioning: Sports Scenes 2
Caption(ep30): a man in a helmet is riding a bmx bike
Caption(ep50): a person rides a dirt bike on a hill
Caption(ep30): a person in a blue shirt is doing a trick in the air on a bike
Caption(ep50): a man is riding a motorcycle on a dirt hill
Caption(ep30): a man doing a trick on a skateboard on a ramp
Caption(ep50): a man doing a skateboard trick on a ramp
Caption(ep30): a skateboarder does a jump on his skateboard
Caption(ep50): a boy in a red shirt does a trick on his skateboard in a skate park
Caption(ep30): two boys are playing soccer in a field
Caption(ep50): a soccer player is playing soccer
Caption(ep30): a group of children playing with a soccer ball in the grass
Caption(ep50): two people are playing soccer on a grassy hill

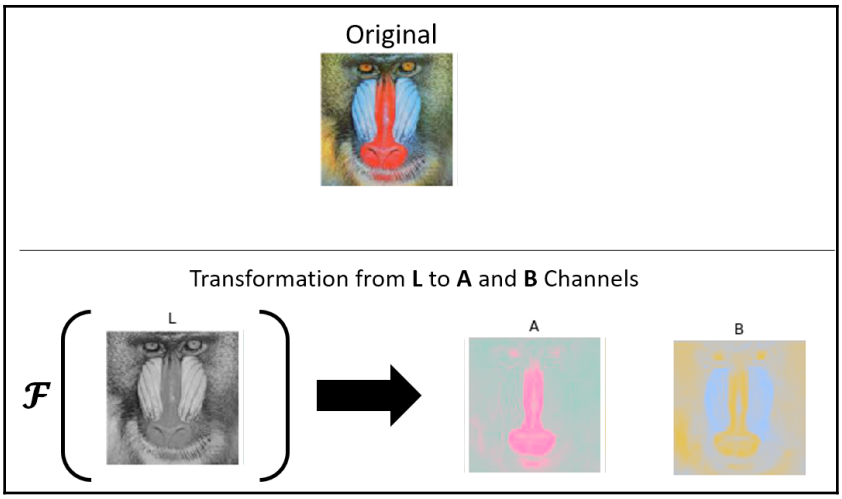
Chapter 12: Image Colorization

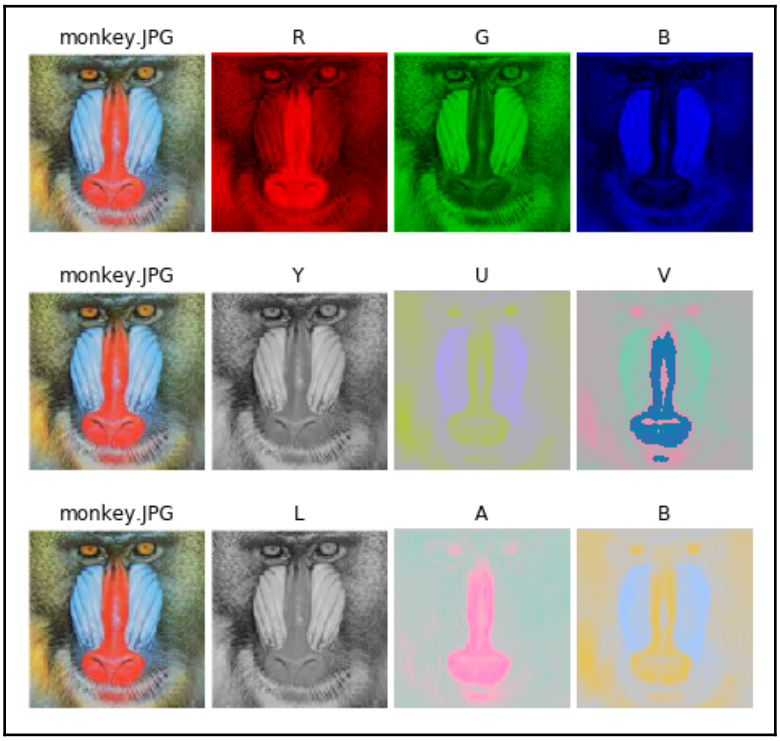


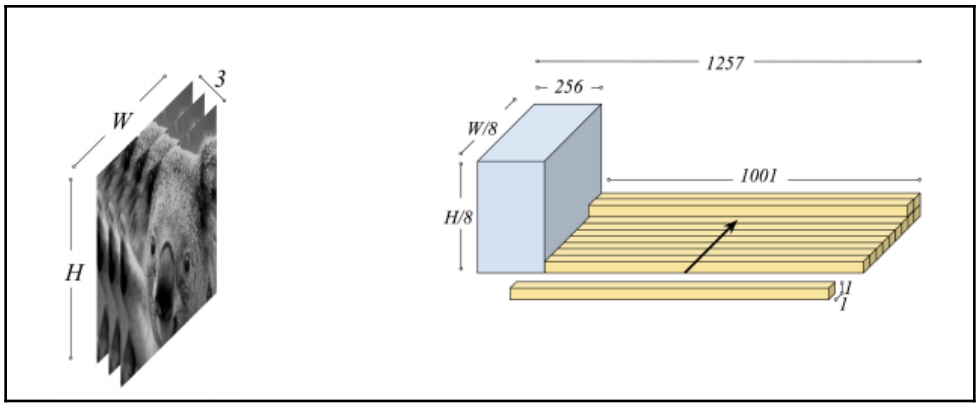


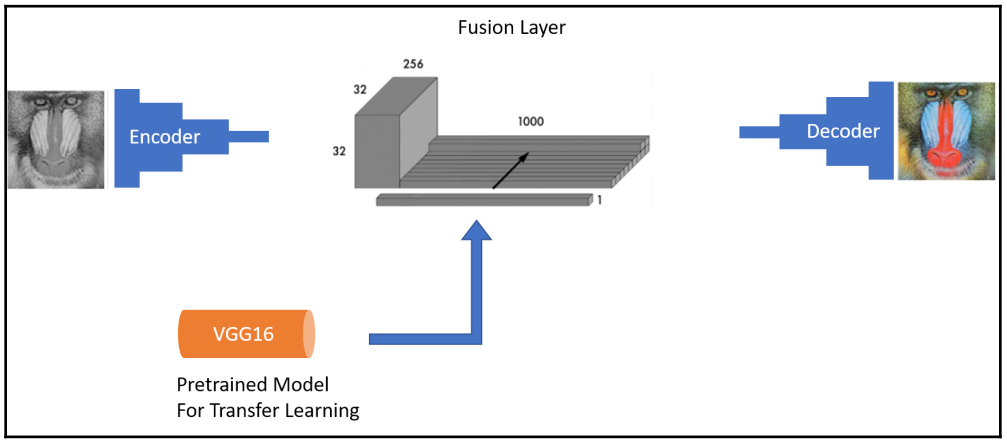

V
+.4
+.3
+.2
+.1
U
−.4 −.3 −.2 −.1 +.1 +.2 +.3 +.4
−.1
−.2
−.3
−.4




Original
Transformation from **L** to **A** and **B** Channels
$\mathcal{F}$
L
A
B


monkey.JPG
R
G
B
monkey.JPG
Y
U
V
monkey.JPG
L
A
B


3
W
H
1257
256
W/8
H/8
1001
1
1


Fusion Layer
Encoder
256
32
32
1000
1
Decoder
VGG16
Pretrained Model
For Transfer Learning


hallucination
grayscale
original
grayscale
hallucination
original
hallucination
grayscale
original