CS4104 Applied Machine Learning

Deep Learning

- Mean average precision
- Cumulative Gain
- Silhouette Index

- Davies Bouldin

- Calinski Harabasz

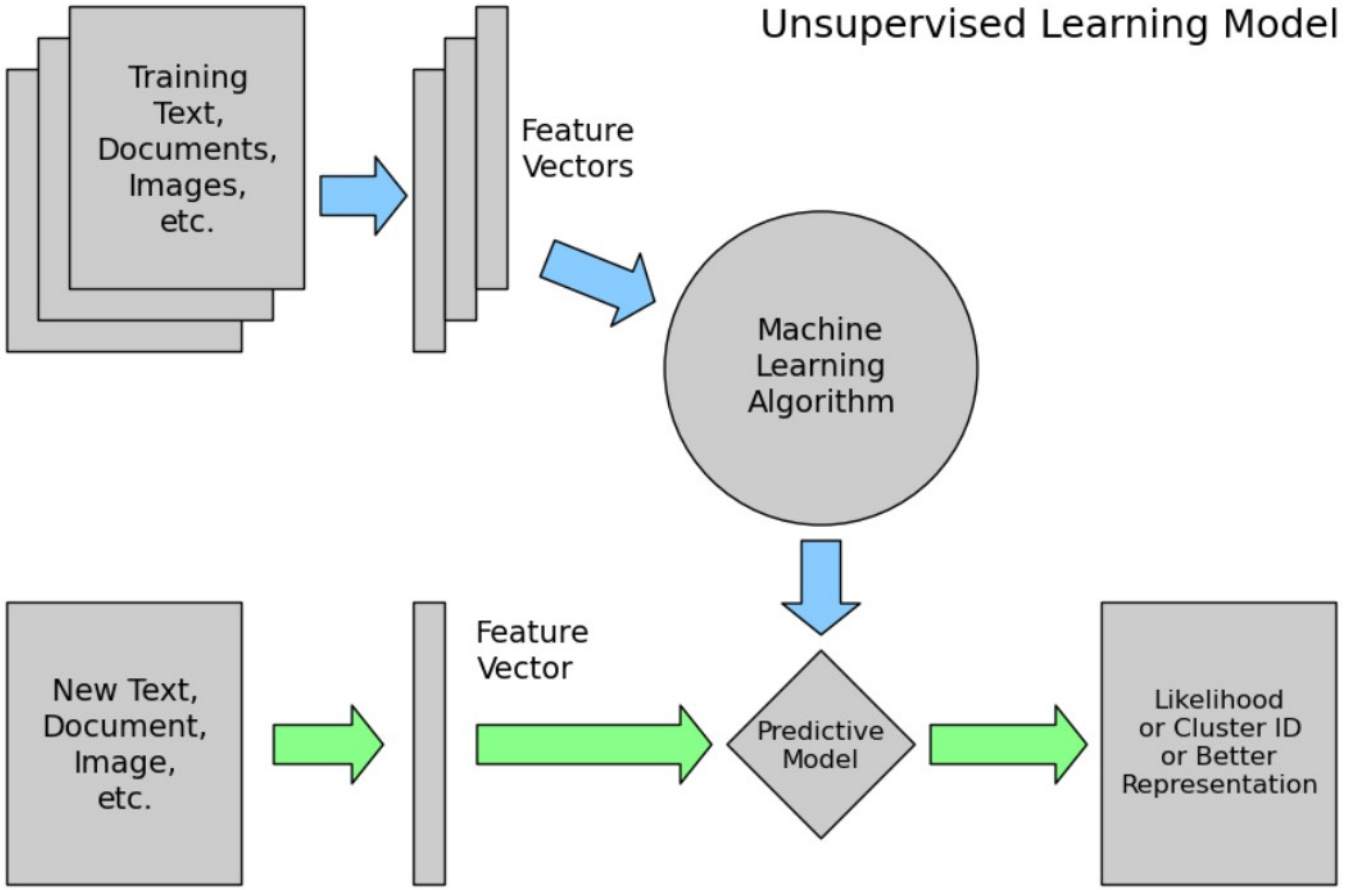
Machine Learning

Field of study that gives computers ability to learn

- Machine learning is programming computers to optimize a performance criterion using example data or past experience.
- There is no need to “learn” to calculate payroll
- Learning is used when:
  - Human expertise does not exist (navigating on Mars),
  - Humans are unable to explain their expertise (speech recognition)
  - Solution changes in time (routing on a computer network)
  - Solution needs to be adapted to particular cases (user biometrics)

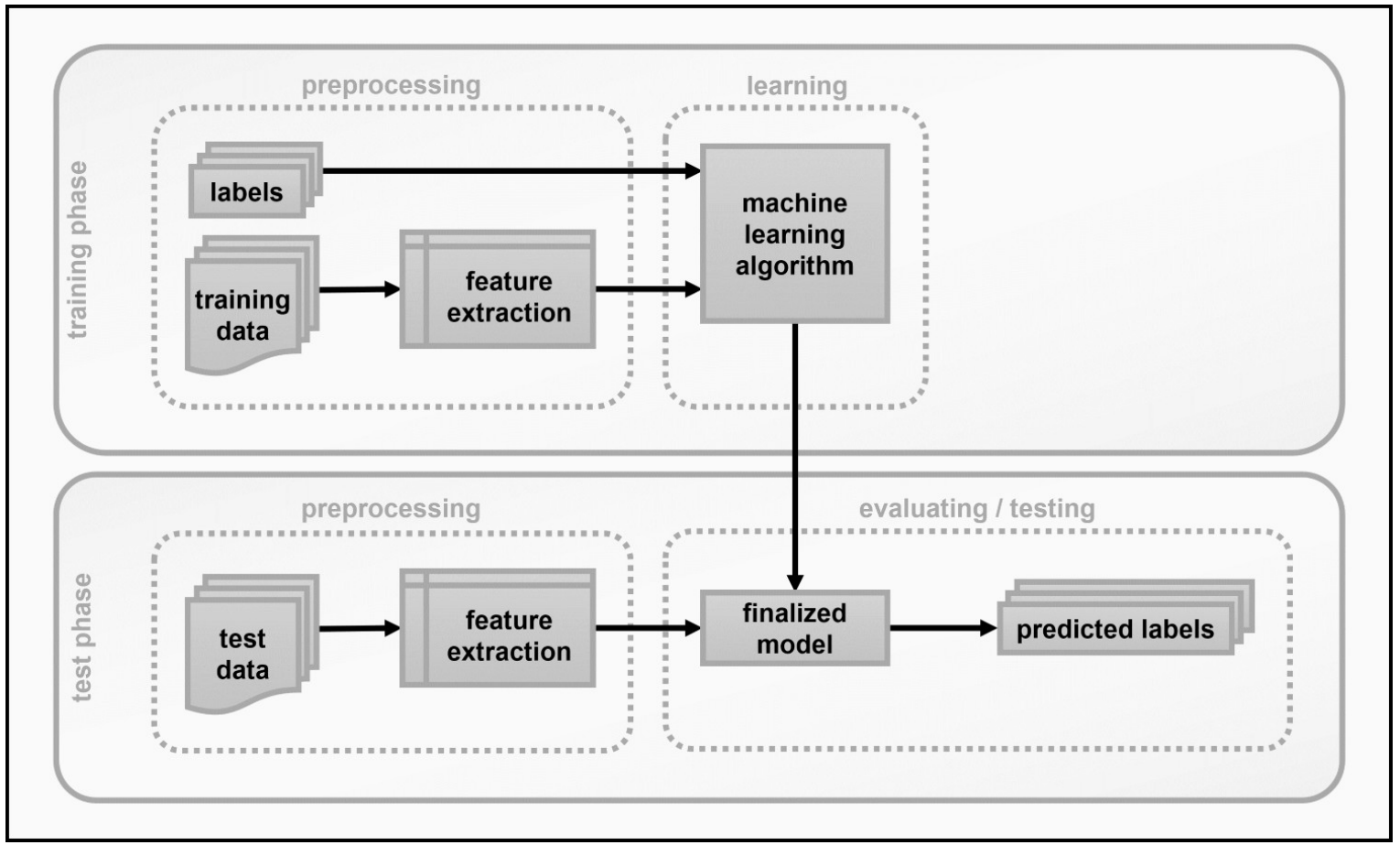
zeshan.khan@nu.edu.pk

Un-Supervised Learning



zeshan.khan@nu.edu.pk

Supervised Learning



zeshan.khan@nu.edu.pk

Image Features

- Color Descriptors
  - Dominant color
  - Color Layout (essentially Primary color on block-by-block basis)
  - Scalable Color (essentially Color histogram)
  - Color Structure (essentially local Color histogram)
  - Color spaces to make things interoperable
- Texture Features
  - Texture Browsing Descriptor - which defines granularity/coarseness, regularity, and direction.
  - Homogeneous Texture Descriptor - which is based on Gabor filter bank
  - Edge Histogram
  - Haralick Features
  - edge detectors
  - corner detectors
  - blob detectors
  - ridge detectors
- Shape Descriptors
  - Region based descriptors are scalar attributes of shape under consideration - such as area, ecentricities etc.
  - Contour based which captures actual characteristic shape features
  - 3D descriptors
- Motion Descriptors for Video
  - Camera Motion (3-D camera motion parameters)
  - Motion Trajectory (of objects in the scene) [e.g. extracted by tracking algorithms] Parametric Motion (e.g. motion vectors, which allows description of motion of scene. But it can be more complex models on various objects).
  - Activity which is more of a semantic descriptor.
- Local Binary and Ternary Patterns

zeshan.khan@nu.edu.pk

Text Features

- Text Vectorization
- Text Frequencies
  - TF
  - IDF
  - TF-IDF
- Bag of Words
- Word Embedding
  - WordtoVec (Google)
  - Global Vectors or GloVe (Stanford)
  - fastText (Facebook)

Learning by examples

- is input
- is target output
- Start with some w and b
  - Change the values of b and w

Gradient Descend

- is the actual out
- is the predicted output
- LR is the learning rate

zeshan.khan@nu.edu.pk

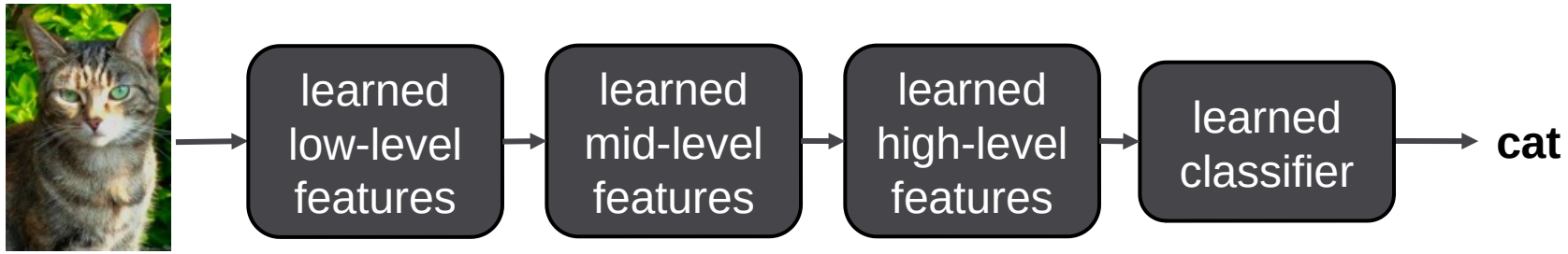
Deep Learning

- Hype: extravagant or intensive publicity or promotion
- Hope: expectation of fulfillment or success
- Yann LeCun quote: DNNs require: “an interplay between intuitive insights, theoretical modeling, practical implementations, empirical studies, and scientific analyses” (Turing Award: 2018)

zeshan.khan@nu.edu.pk

“Traditional” machine learning:

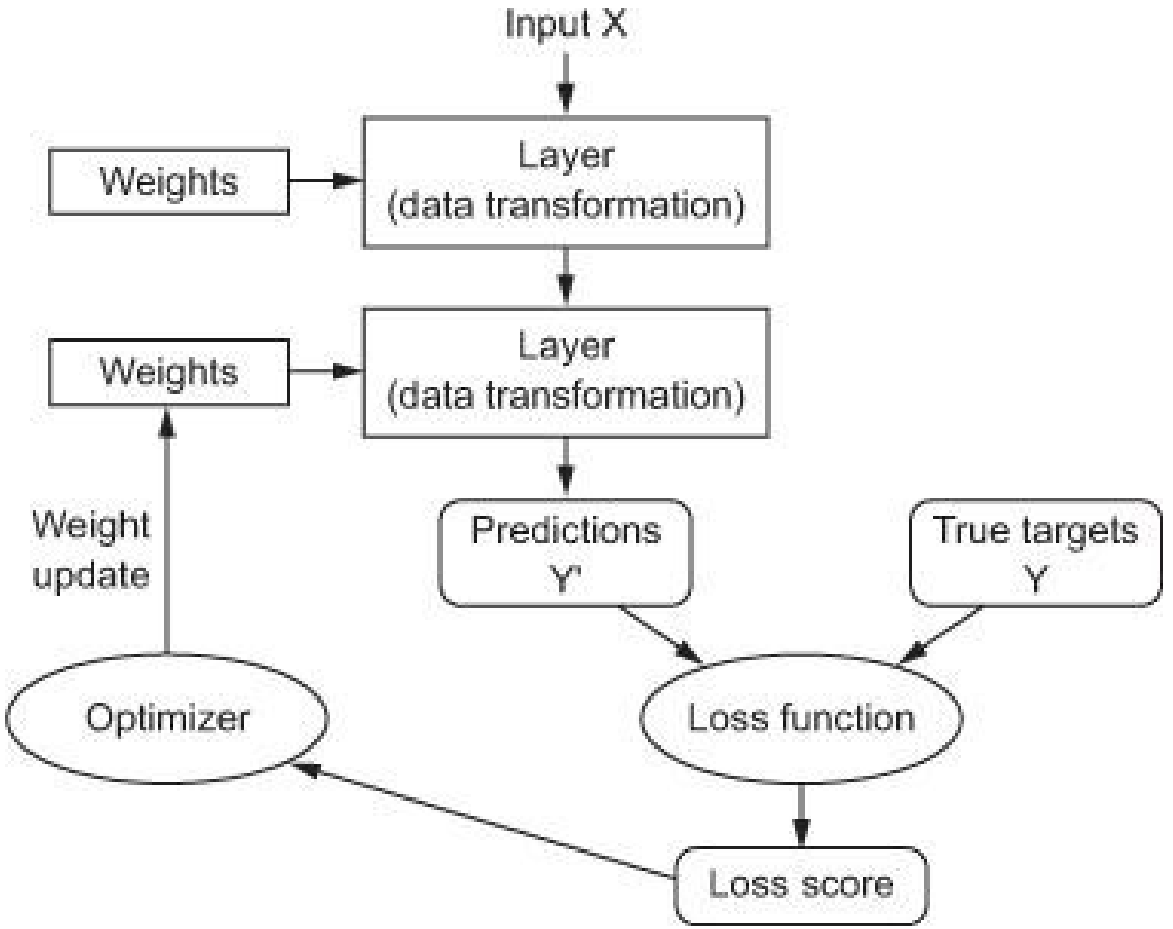
Deep, “end-to-end” learning:



zeshan.khan@nu.edu.pk

Anatomy of a deep neural network

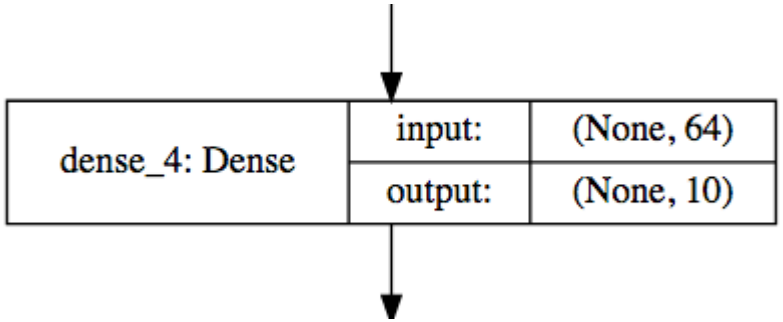
- Layers
- Input data and targets
- Loss function
- Optimizer



zeshan.khan@nu.edu.pk

Layers

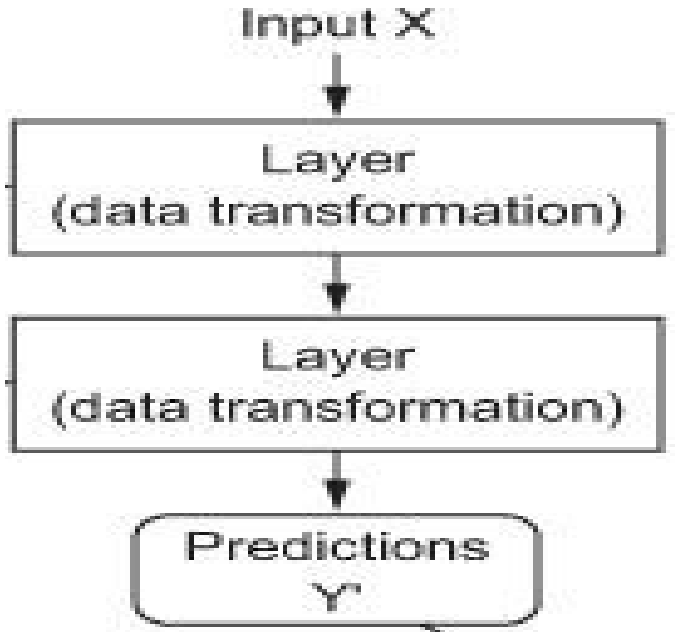
- Data processing modules
- Many different kinds exist
  - densely connected
  - Convolutional
  - Recurrent
  - pooling, flattening, merging, normalization
- Input: one or more tensors
  output: one or more tensors
- Usually have a state, encoded as weights
  - learned, initially random
- When combined, form a network or
  a model

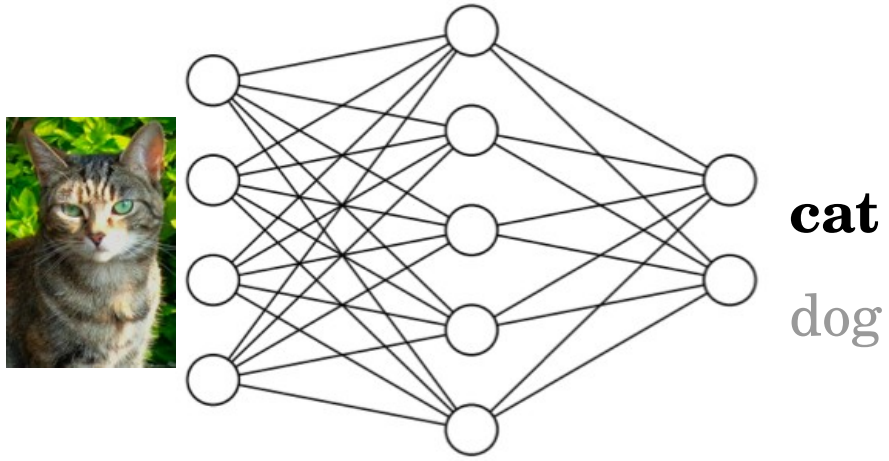


zeshan.khan@nu.edu.pk

Input data and targets

- The network maps the input data X to predictions Y′
- During training, the predictions Y′ are compared to true targets Y using the loss function





zeshan.khan@nu.edu.pk

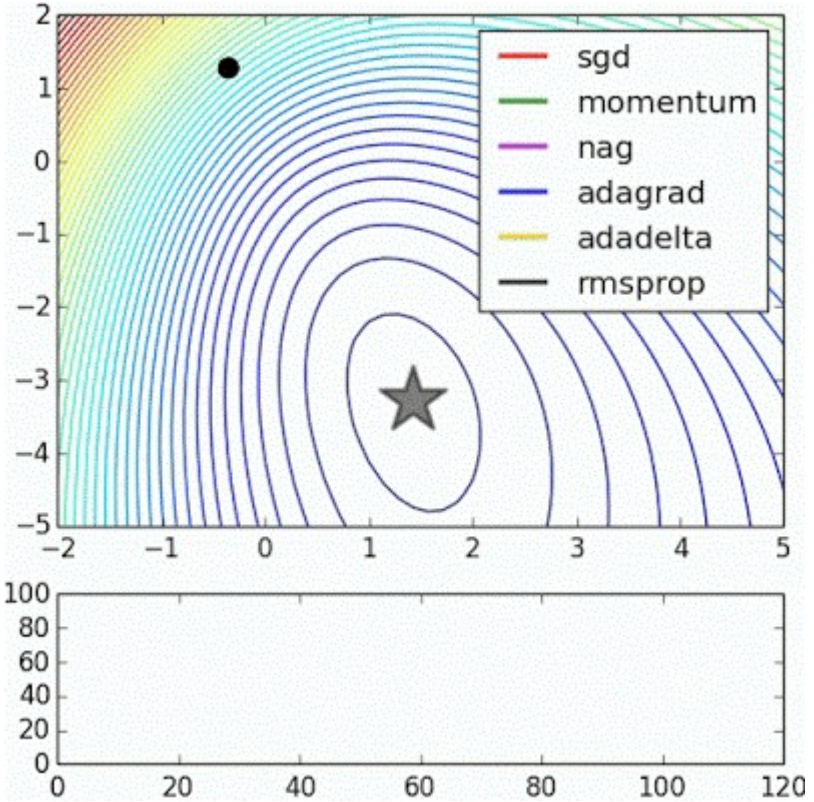
Loss function

- The quantity to be minimized (optimized) during training
  - the only thing the network cares about
  - there might also be other metrics you care about
- Common tasks have “standard” loss functions:
  - *mean squared error* for regression
  - *binary cross-entropy* for two-class classification
  - *categorical cross-entropy* for multi-class classification

zeshan.khan@nu.edu.pk

Optimizer

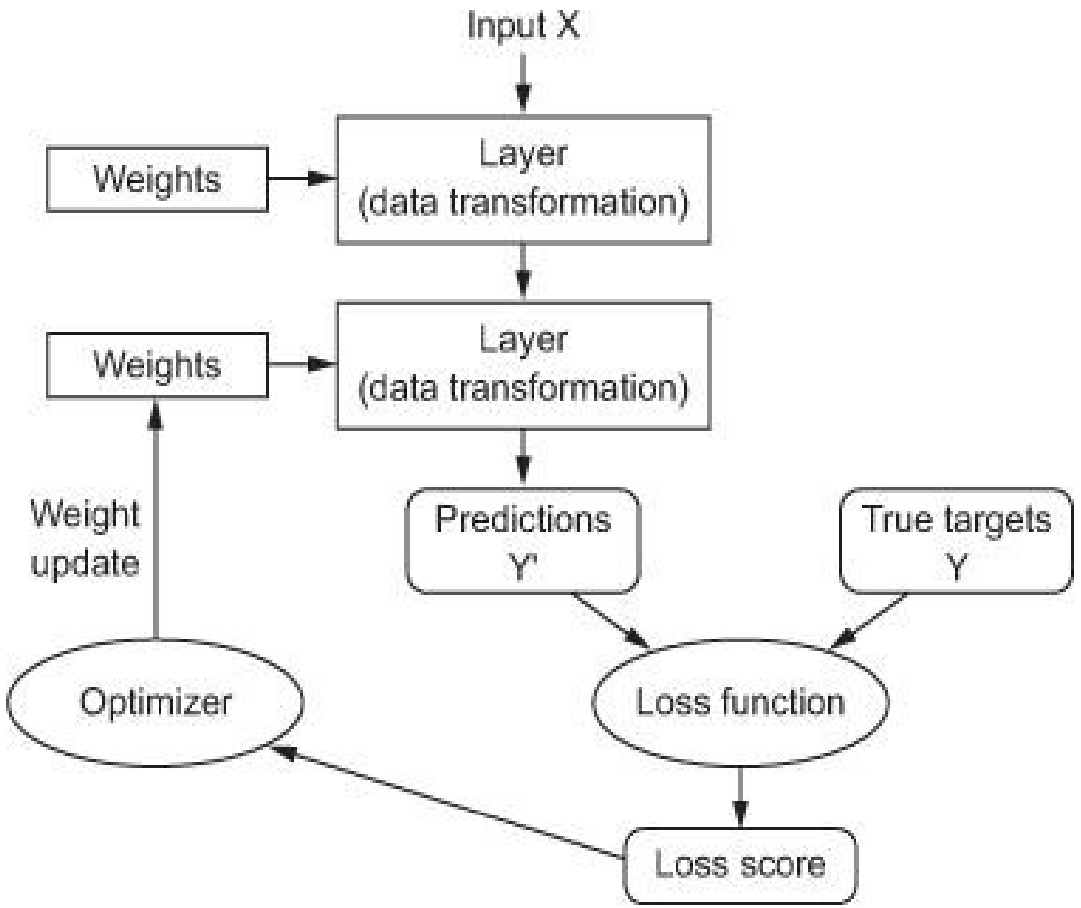
- How to update the weights based on the loss function
- *Learning rate (+scheduling)*
- Stochastic gradient descent, momentum, and their variants
  - RMSProp is usually a good first choice



Animation from:
https://imgur.com/s25RsOr

zeshan.khan@nu.edu.pk

Anatomy of a deep neural network



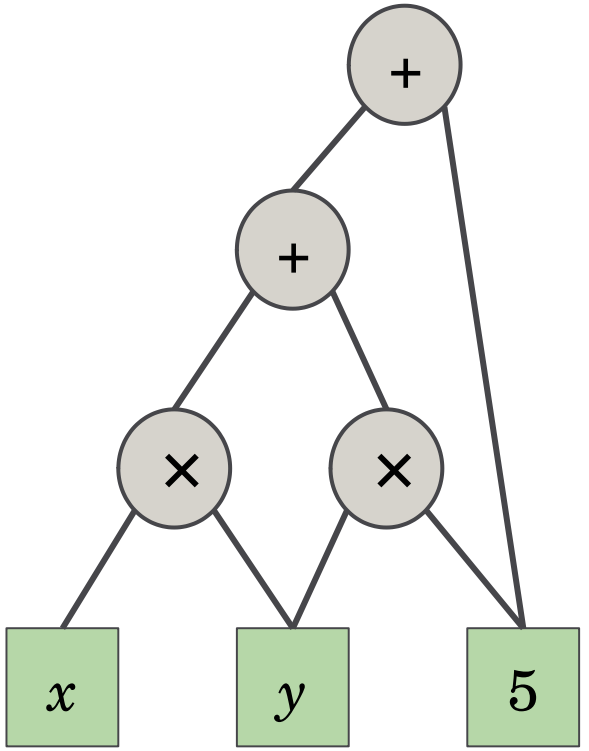
zeshan.khan@nu.edu.pk

Frameworks for DL

zeshan.khan@nu.edu.pk

Deep learning frameworks

- Actually tools for defining static or dynamic general-purpose computational graphs
- Automatic differentiation
- Seamless CPU / GPU usage
  - multi-GPU, distributed
- Python/numpy or R
- Open source



$$xy + 5y + 5$$

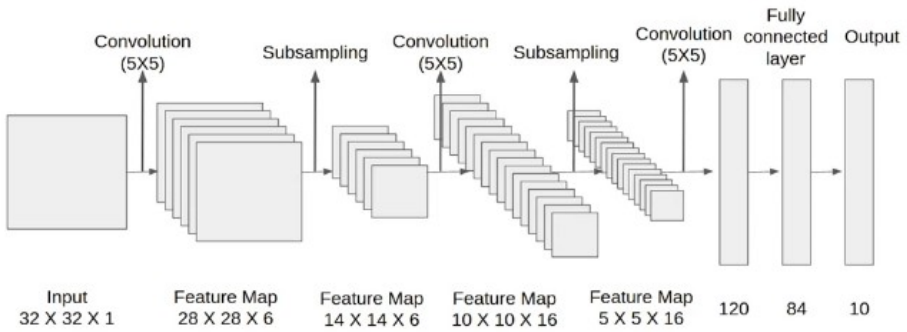
zeshan.khan@nu.edu.pk

CS4104 Applied Machine Learning

Neural Networks with Timeline

LeNet (Yann LeCun 1989)

- Backpropagation Applied to Handwritten Zip Code Recognition
- Subsampling/Max Pooling
- Tested on
  - ZipCode
  - MNIST (0 to 9 digits)
- From 2 Convolutions to 5 Convolutions

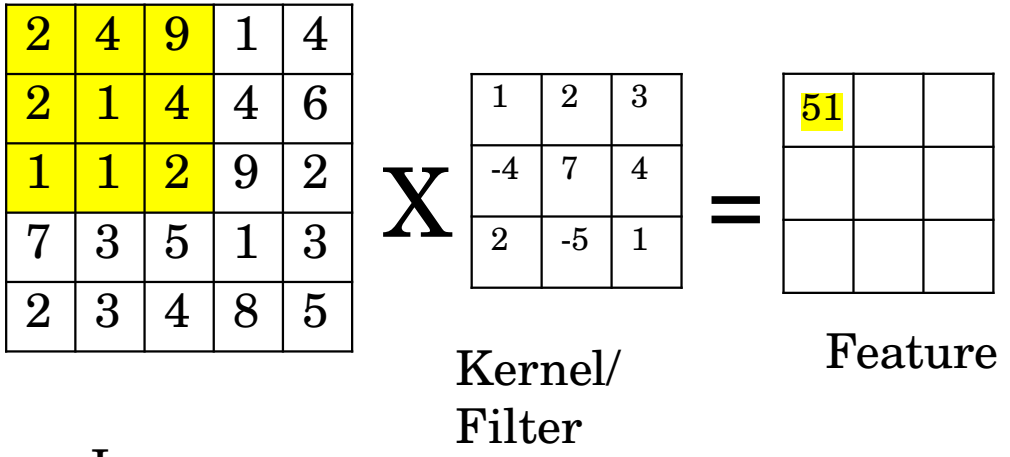


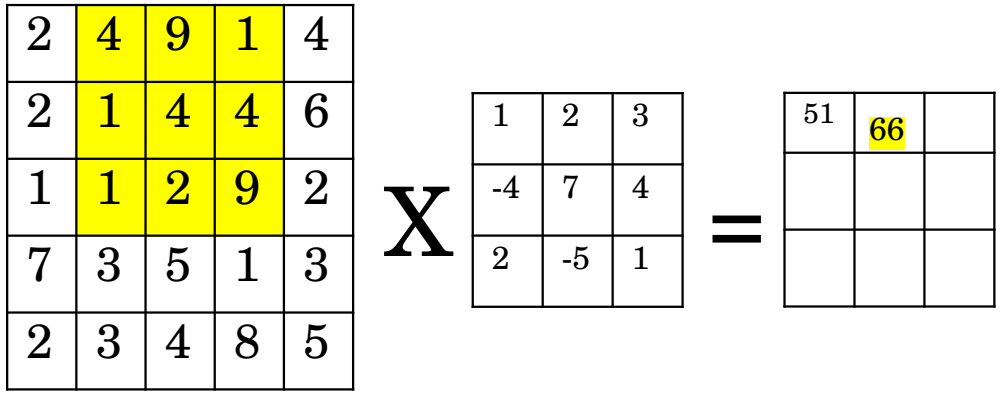
zeshan.khan@nu.edu.pk

Convolution

- For r in Rows:
  - For c in columns:
    - Res[r,c]=sum(img[r,c] *kernel)

- 2*1+4*2+9*3=37
- 2*-4+1*7+4*4=15
- 1*2+1*-5+2*1=-1
- Total=51

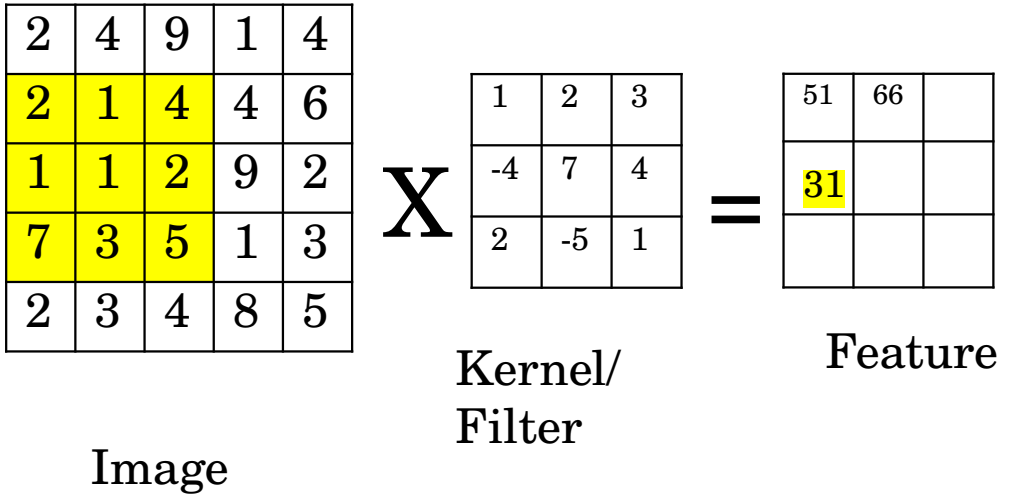


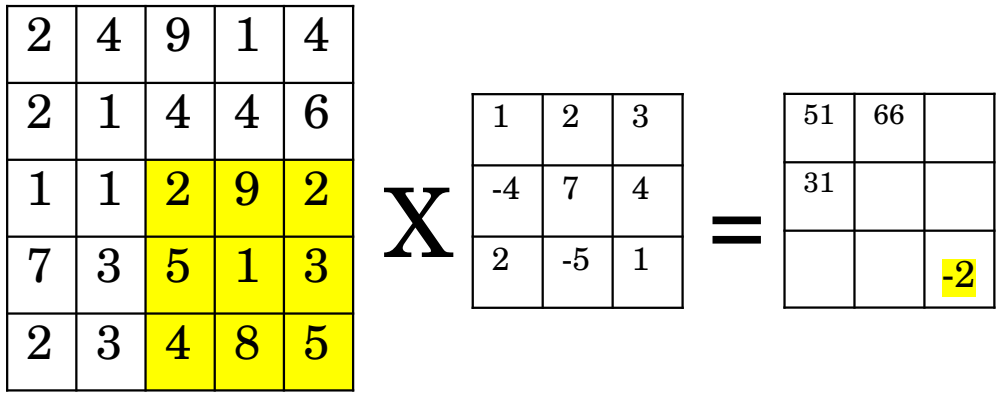


zeshan.khan@nu.edu.pk

Convolution

- For r in Rows:
  - For c in columns:
    - Res[r,c]=sum(img[r,c] *kernel)

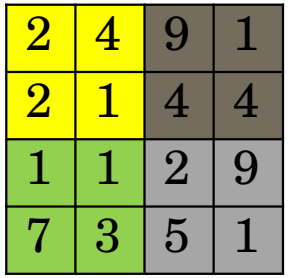




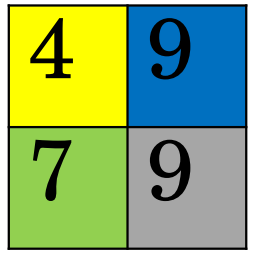
zeshan.khan@nu.edu.pk

Pooling

- Combine pixels into a single pixel with
  - Max
  - Min
  - Average



Image/Source

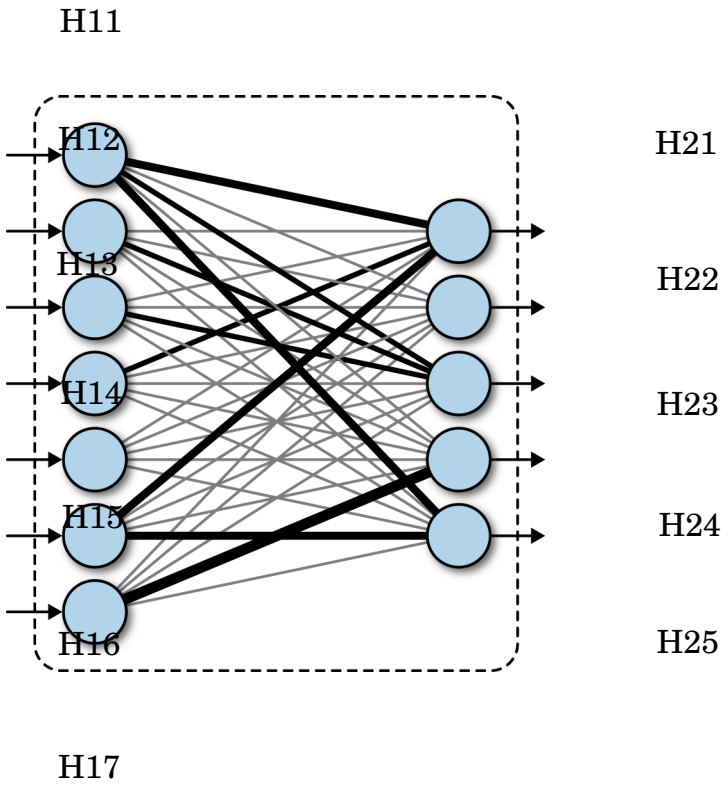


Max Pooling (2*2)

zeshan.khan@nu.edu.pk

Fully Connected Layer

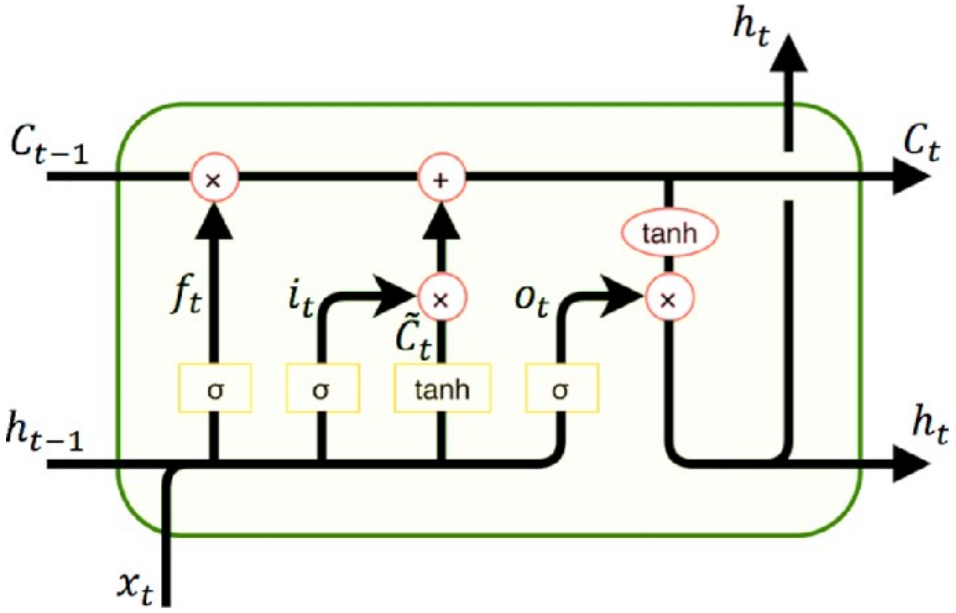
- …



zeshan.khan@nu.edu.pk

Recurrent Nets: LSTMs (1997)

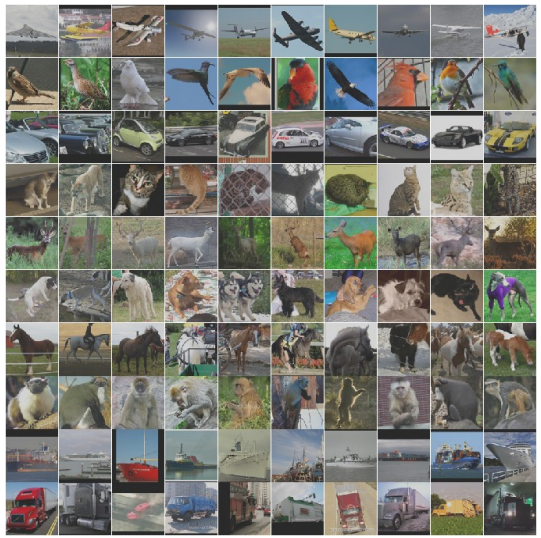
- Speech Recognition
- Long Short-term Memory

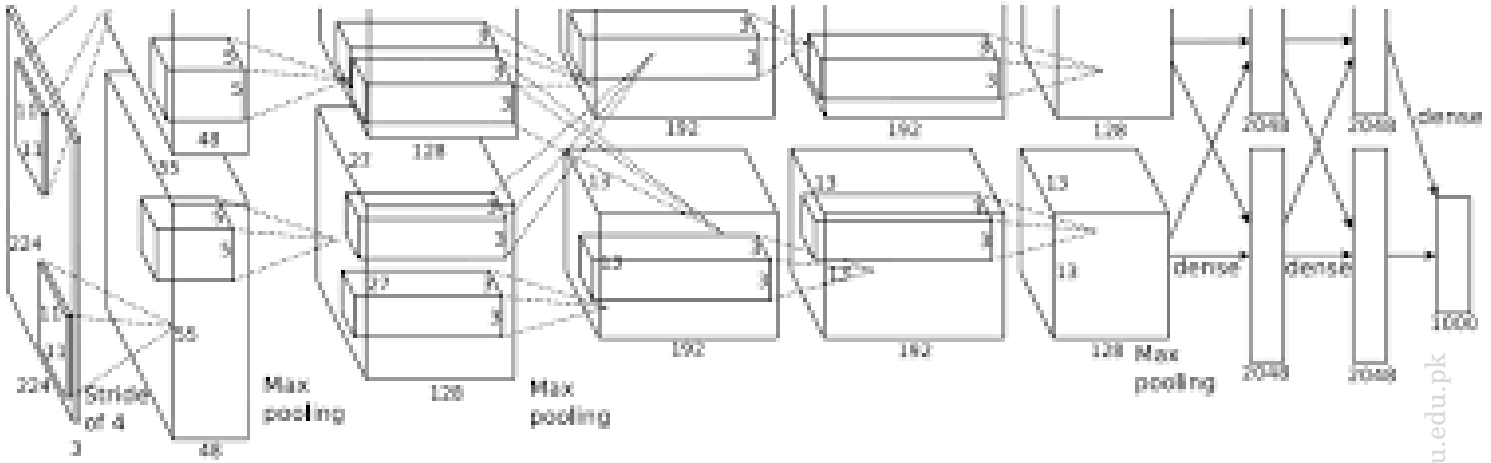


zeshan.khan@nu.edu.pk

Convolutional NNs

- AlexNet (2012)
- ImageNet
- Dataset Size 200 GB





zeshan.khan@nu.edu.pk

VGG (Visual Geometry Group)

- Very Deep Convolutional Networks for Large-Scale Image Recognition.
- VGG16 (16 Convolution Layer)
- VGG19 (19 Convolution Layer)
- Applied on ImageNet dataset
- Rectified Linear Unit (ReLU)
- Softmax

```
from keras.applications.vgg19 import VGG19
VGG19().summary()
```

- Convolution Layers
  - Convolution2D
  - ReLU
  - Max Pooling2D
- Fully Connected Layers
  - Dense (4096)
  - ReLU
  - Dense (4096)
  - ReLU
  - Dense (1000)
  - Softmax
- Input (224*224*3)
- Output (1000)
- Parameters: 143,667,240 (VGG19) [Mostly at FC]

zeshan.khan@nu.edu.pk

Parameters

block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 1000)	4097000

Total params: 143,667,240
Trainable params: 143,667,240
Non-trainable params: 0

zeshan.khan@nu.edu.pk

ResNet

- Kaiming He (2015)
- ILSVRC 2015 classification competition (ImageNet)
- MS COCO segmentation in the ILSVRC & COCO competitions of 2015
- ResNet
  - 34 Layered
  - 50 Layered
  - 101 Layered
  - 152 Layered

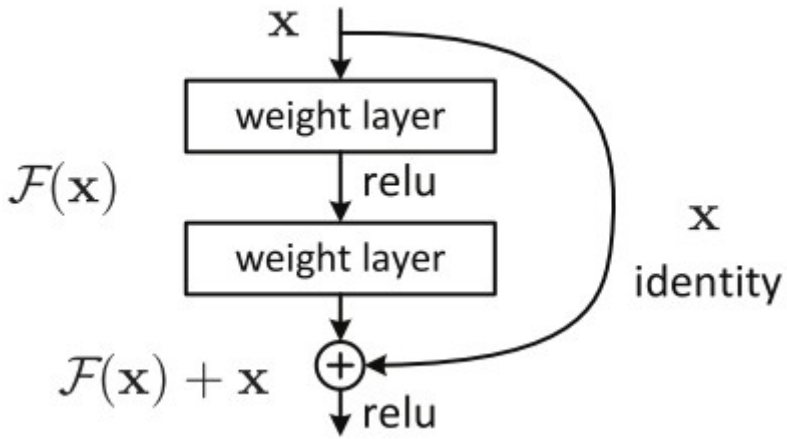
- Input:
  - 224*224*3
- Convolution
- Max Pooling
- Batch Normalization

- Fully Connected (2048)
- Fully Connected (1000)
- Parameters (25,636,712) < Parameters of VGG

zeshan.khan@nu.edu.pk

Residual Block

- vanishing gradient
  - nth Derivative can become zero in floating point range

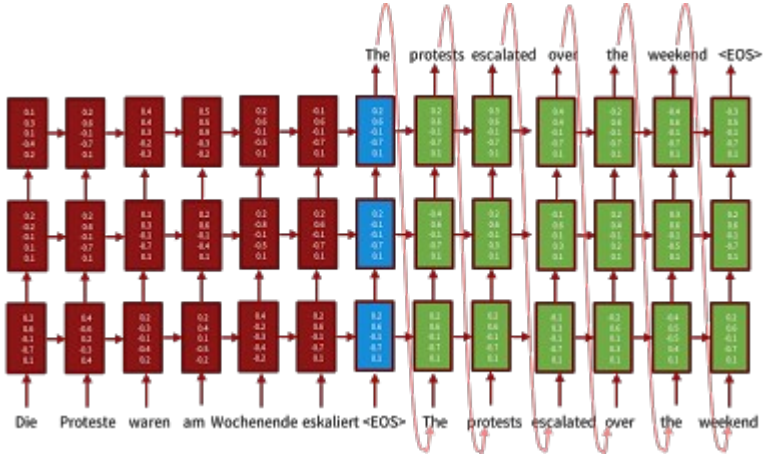


Source: Towards data science

zeshan.khan@nu.edu.pk

Language Translation (2016)

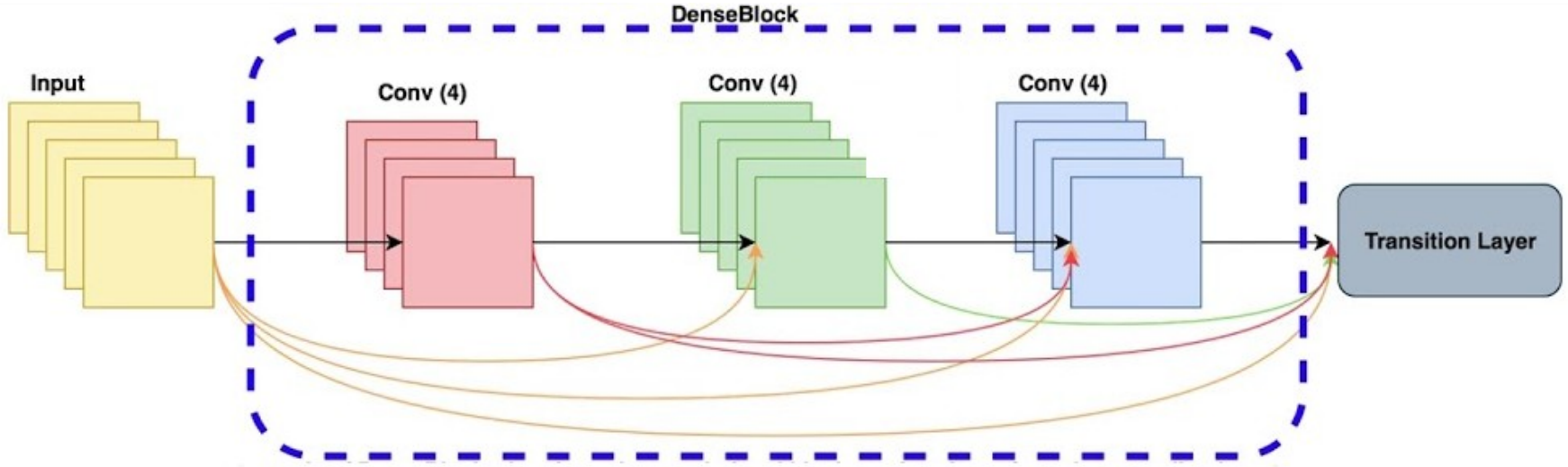
Sequence-to-sequence models with
LSTMs and attention



zeshan.khan@nu.edu.pk

DenseNet

- G Huang (2017)
- NetWork of the Networks



zeshan.khan@nu.edu.pk

Image Classification DL Models

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	0.790	0.945	22,910,480	126	109.42	8.06
VGG16	528	0.713	0.901	138,357,544	23	69.50	4.16
VGG19	549	0.713	0.900	143,667,240	26	84.75	4.38
ResNet50	98	0.749	0.921	25,636,712	-	58.20	4.55
ResNet101	171	0.764	0.928	44,707,176	-	89.59	5.19
ResNet152	232	0.766	0.931	60,419,944	-	127.43	6.54
ResNet50V2	98	0.760	0.930	25,613,800	-	45.63	4.42
ResNet101V2	171	0.772	0.938	44,675,560	-	72.73	5.43
ResNet152V2	232	0.780	0.942	60,380,648	-	107.50	6.64

zeshan.khan@nu.edu.pk

Image Classification DL Models

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
InceptionResNetV2	215	0.803	0.953	55,873,736	572	130.19	10.02
MobileNet	16	0.704	0.895	4,253,864	88	22.60	3.44
MobileNetV2	14	0.713	0.901	3,538,984	88	25.90	3.83
DenseNet121	33	0.750	0.923	8,062,504	121	77.14	5.38
DenseNet169	57	0.762	0.932	14,307,880	169	96.40	6.28
DenseNet201	80	0.773	0.936	20,242,984	201	127.24	6.67
NASNetMobile	23	0.744	0.919	5,326,716	-	27.04	6.70
NASNetLarge	343	0.825	0.960	88,949,818	-	344.51	19.96

zeshan.khan@nu.edu.pk

Image Classification DL Models

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
EfficientNetB0	29	-	-	5,330,571	-	46.00	4.91
EfficientNetB1	31	-	-	7,856,239	-	60.20	5.55
EfficientNetB2	36	-	-	9,177,569	-	80.79	6.50
EfficientNetB3	48	-	-	12,320,535	-	139.97	8.77
EfficientNetB4	75	-	-	19,466,823	-	308.33	15.12
EfficientNetB5	118	-	-	30,562,527	-	579.18	25.29
EfficientNetB6	166	-	-	43,265,143	-	958.12	40.45
EfficientNetB7	256	-	-	66,658,687	-	1578.90	61.62

zeshan.khan@nu.edu.pk