Deep Learning

By MXK

Summary

1- Machine Learning Roadmap

2- Deep Learning Roadmap

3- Convolutional Neural Network

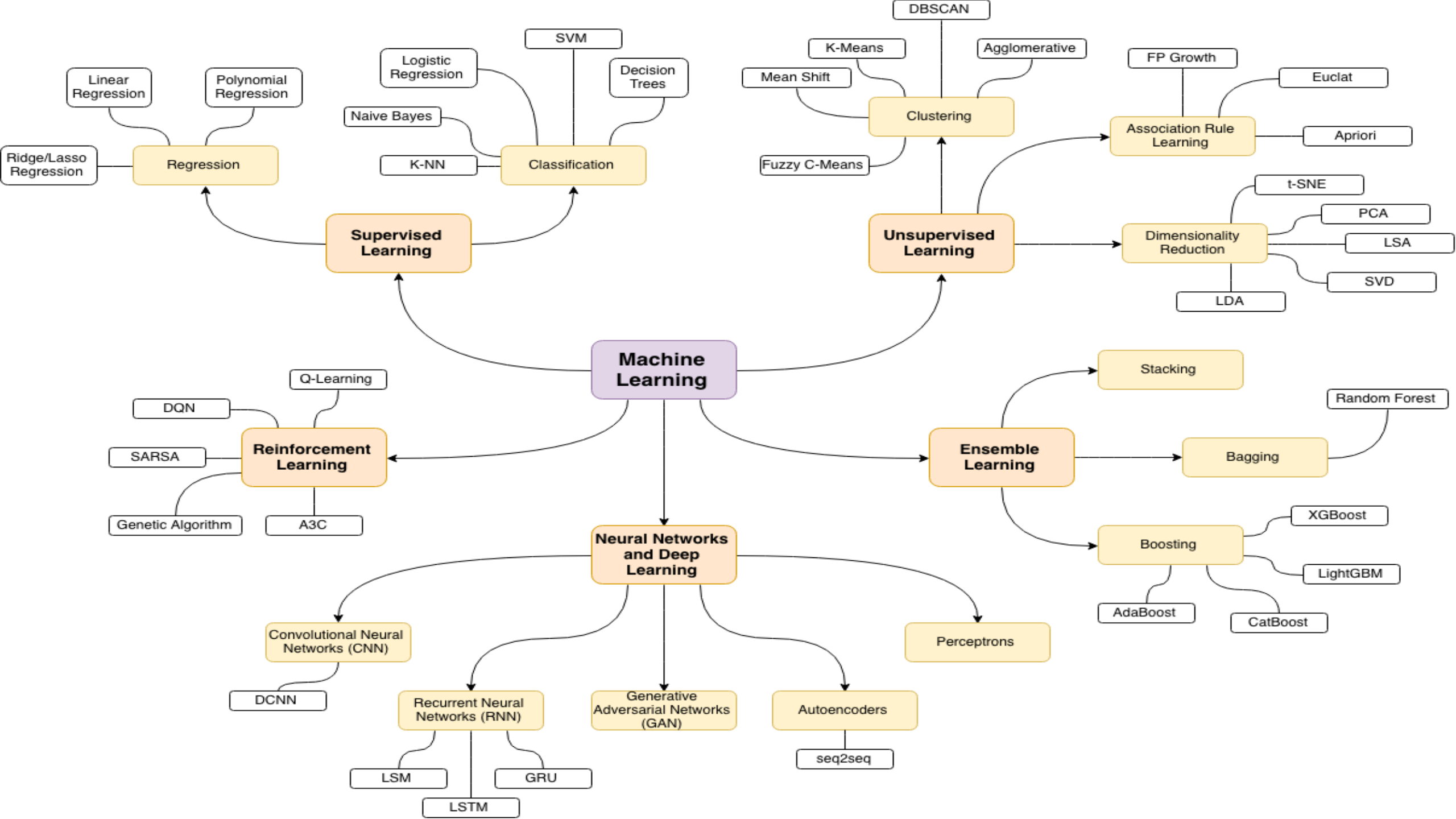
4- Recurrent Neural Network

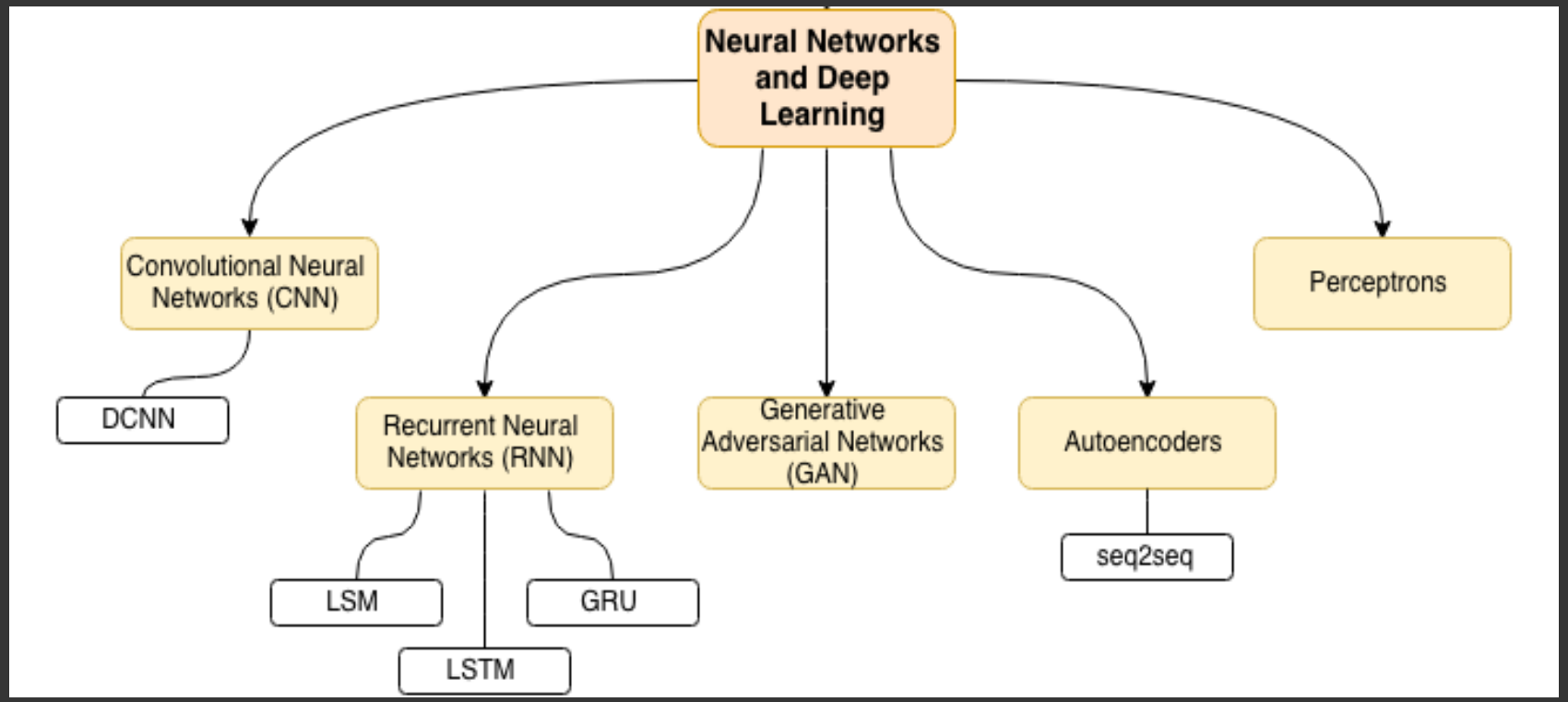
5- Long Short Term Memory

6- Microsoft Azure Example

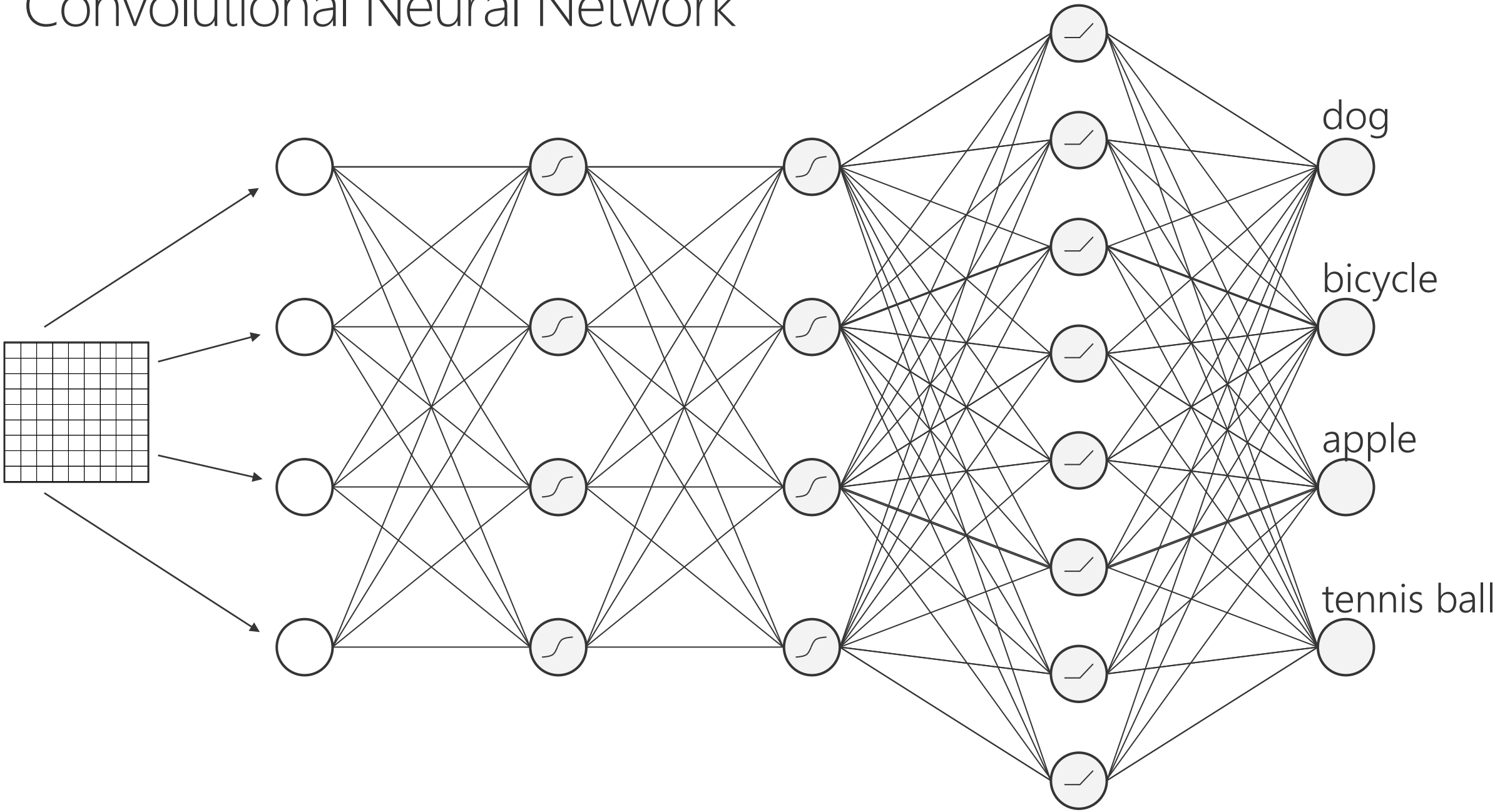
7- Transfer Learning

8- Closing Thoughts

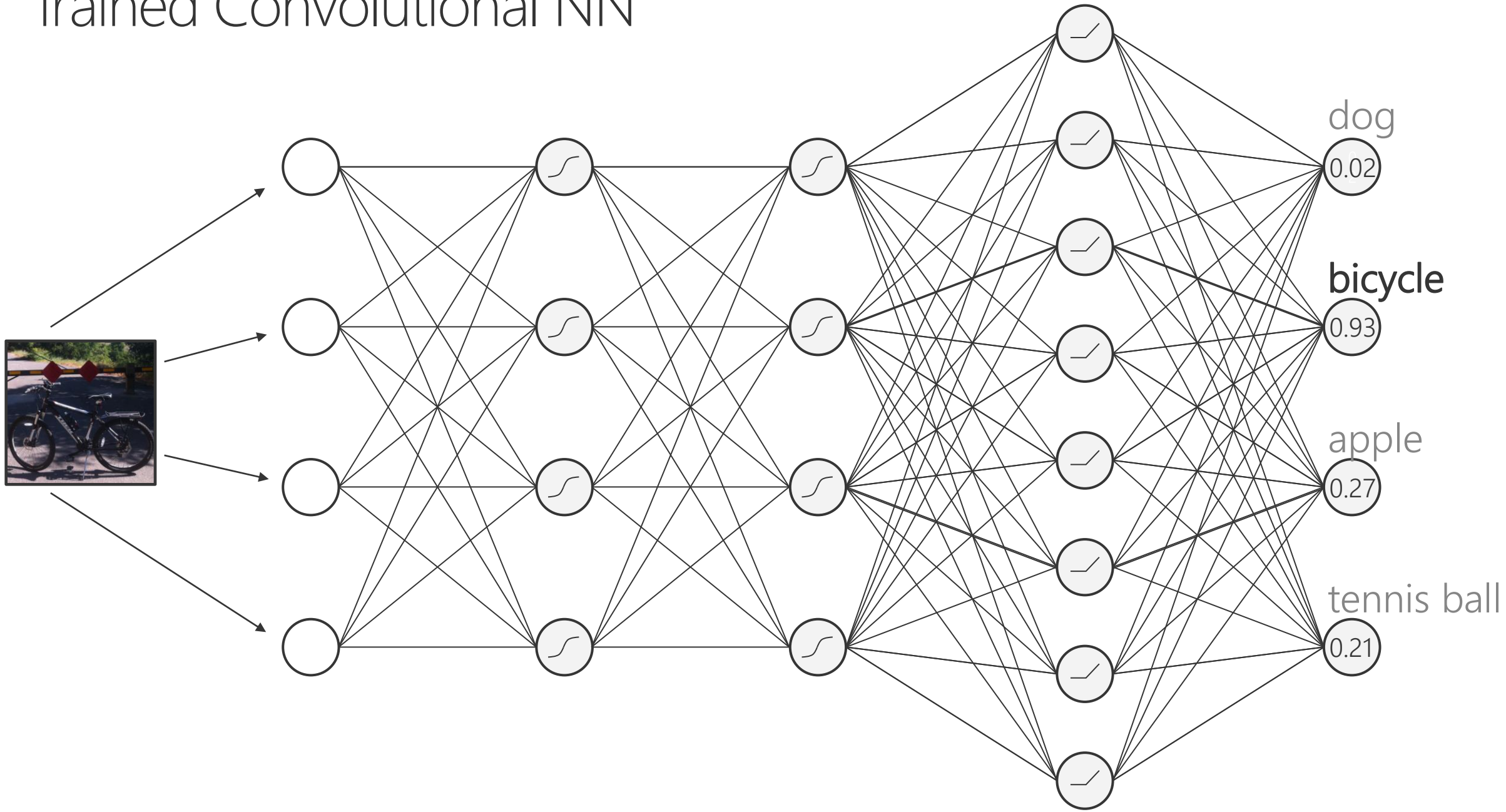

Machine Learning
Supervised Learning
Regression
Linear Regression
Polynomial Regression
Ridge/Lasso Regression
Classification
SVM
Logistic Regression
Decision Trees
Naive Bayes
K-NN
Unsupervised Learning
Clustering
DBSCAN
K-Means
Agglomerative
Mean Shift
Fuzzy C-Means
Association Rule Learning
FP Growth
Euclat
Apriori
Dimensionality Reduction
t-SNE
PCA
LSA
SVD
LDA
Reinforcement Learning
Q-Learning
DQN
SARSA
Genetic Algorithm
A3C
Ensemble Learning
Stacking
Bagging
Random Forest
Boosting
XGBoost
LightGBM
AdaBoost
CatBoost
Neural Networks and Deep Learning
Convolutional Neural Networks (CNN)
DCNN
Recurrent Neural Networks (RNN)
LSM
GRU
LSTM
Generative Adversarial Networks (GAN)
Autoencoders
seq2seq
Perceptrons


Neural Networks and Deep Learning
Convolutional Neural Networks (CNN)
DCNN
Recurrent Neural Networks (RNN)
LSM
GRU
LSTM
Generative Adversarial Networks (GAN)
Autoencoders
seq2seq
Perceptrons

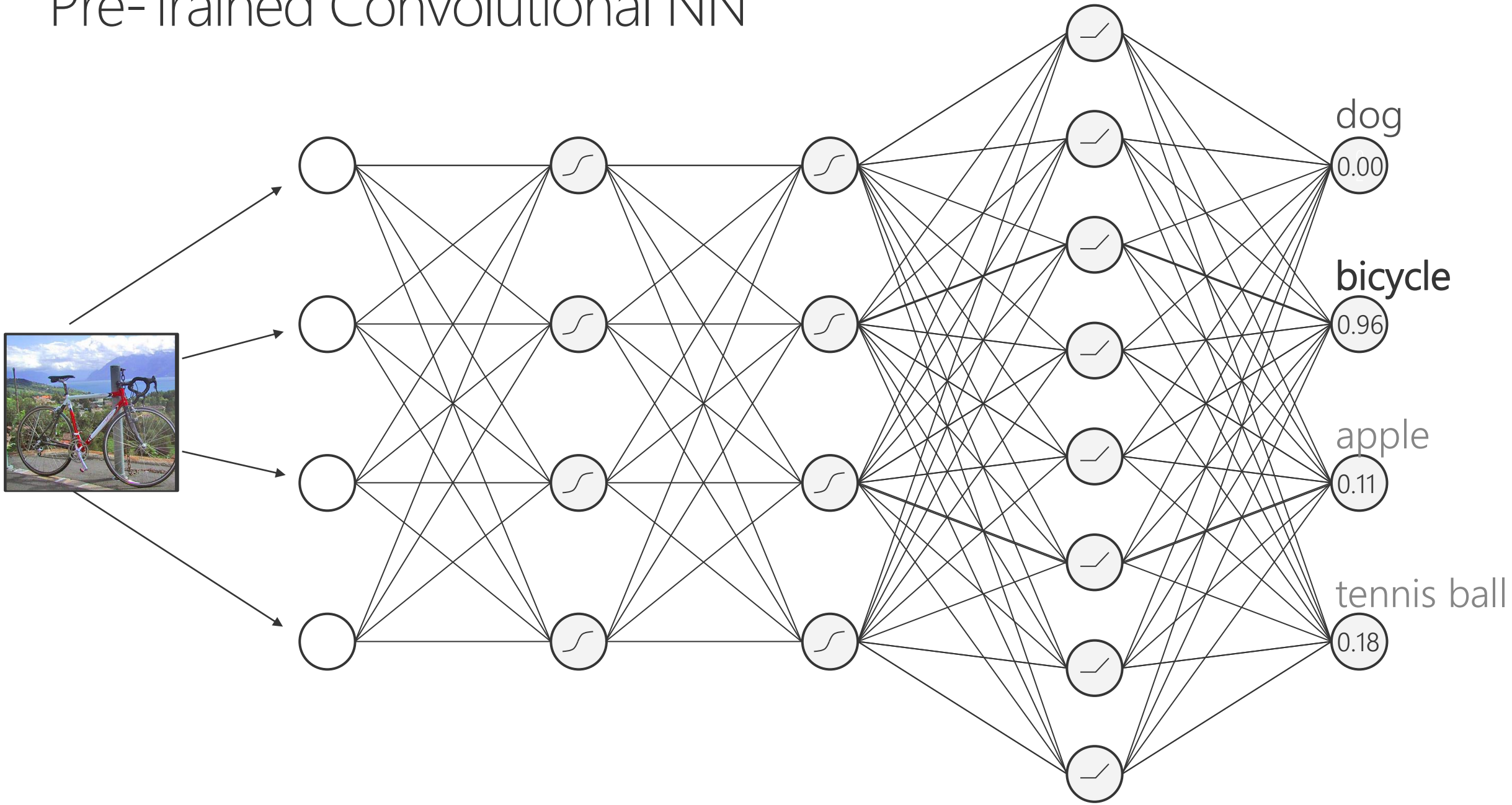
Convolutional Neural Network



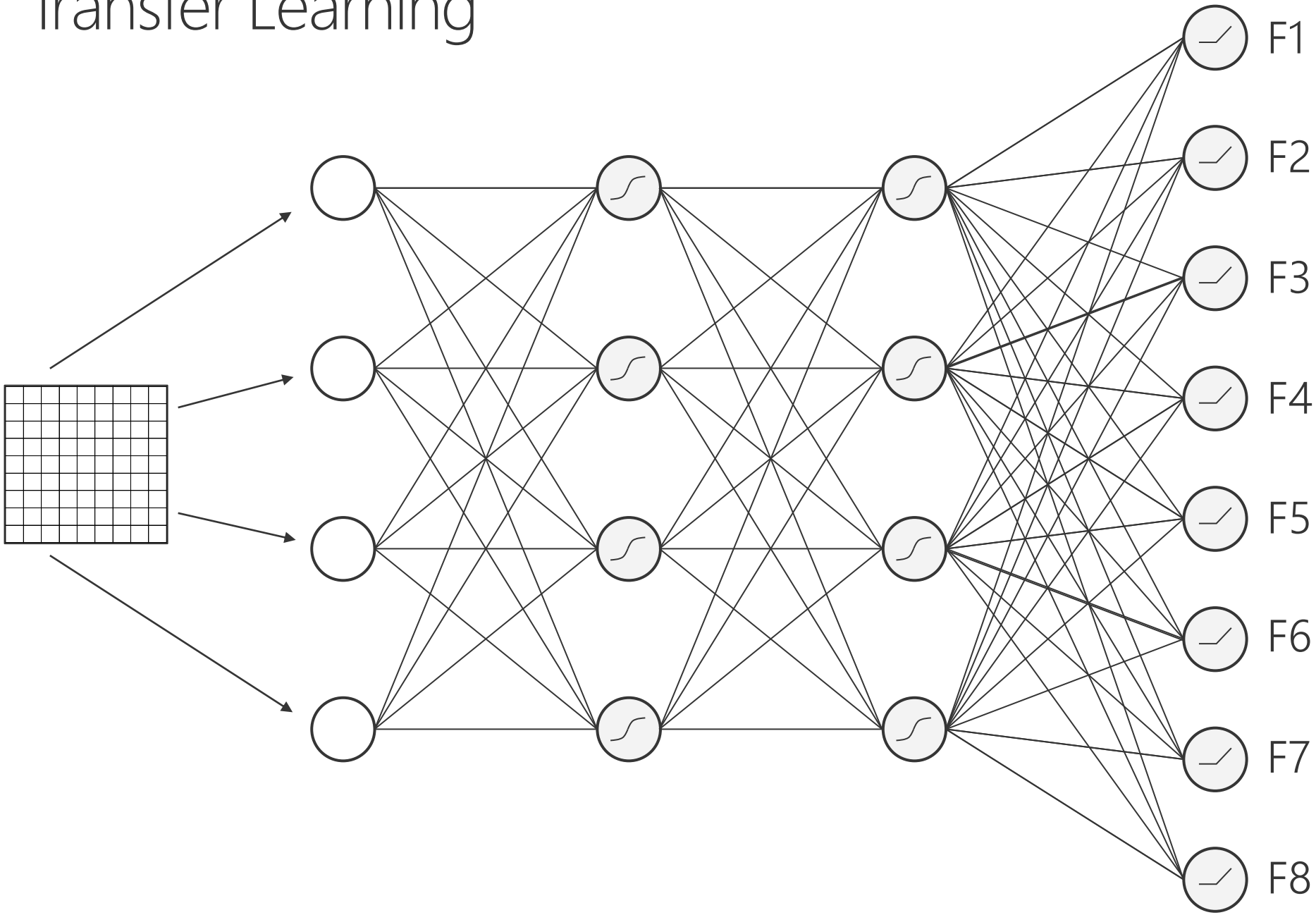
Trained Convolutional NN



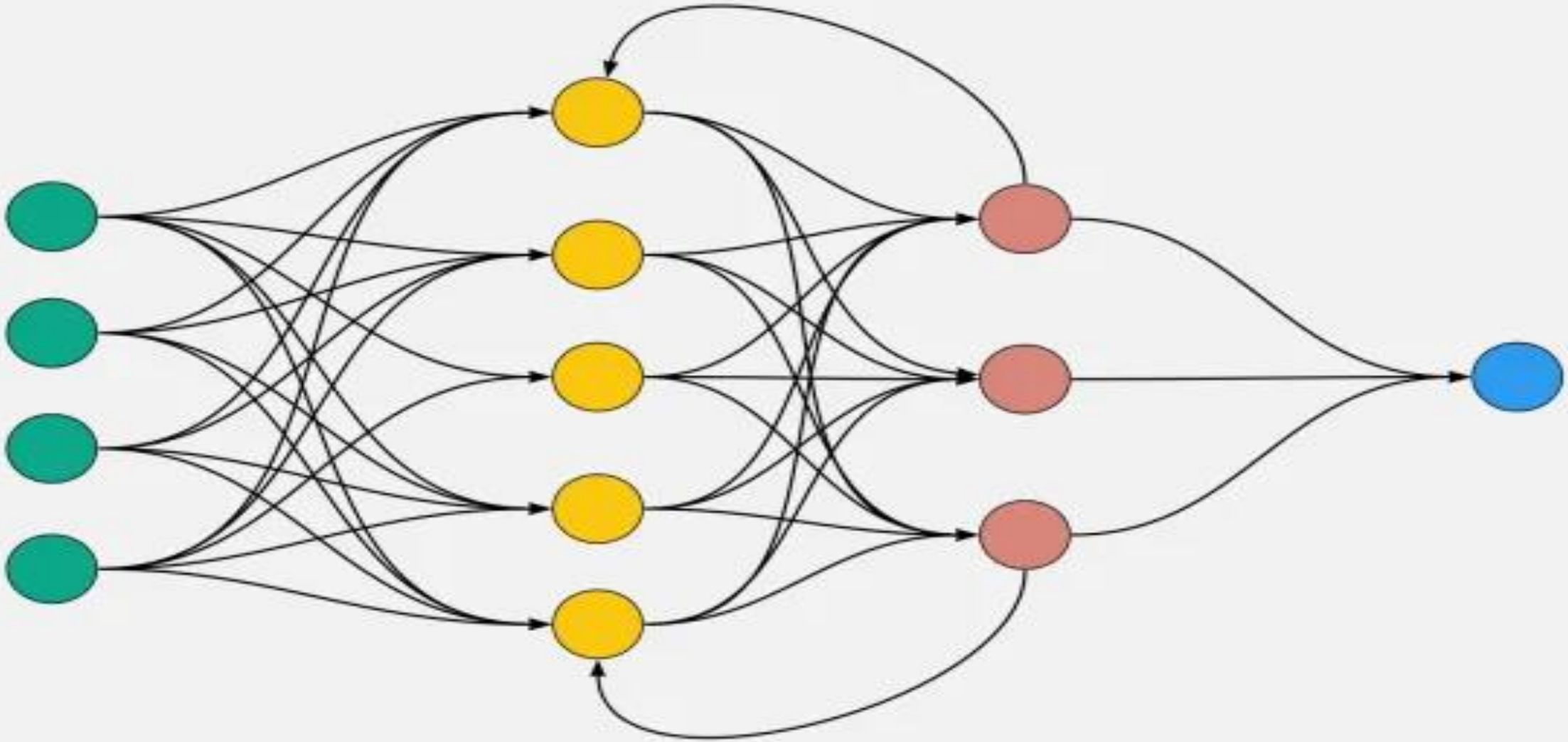
Pre-Trained Convolutional NN



Transfer Learning



Recurrent Neural Network



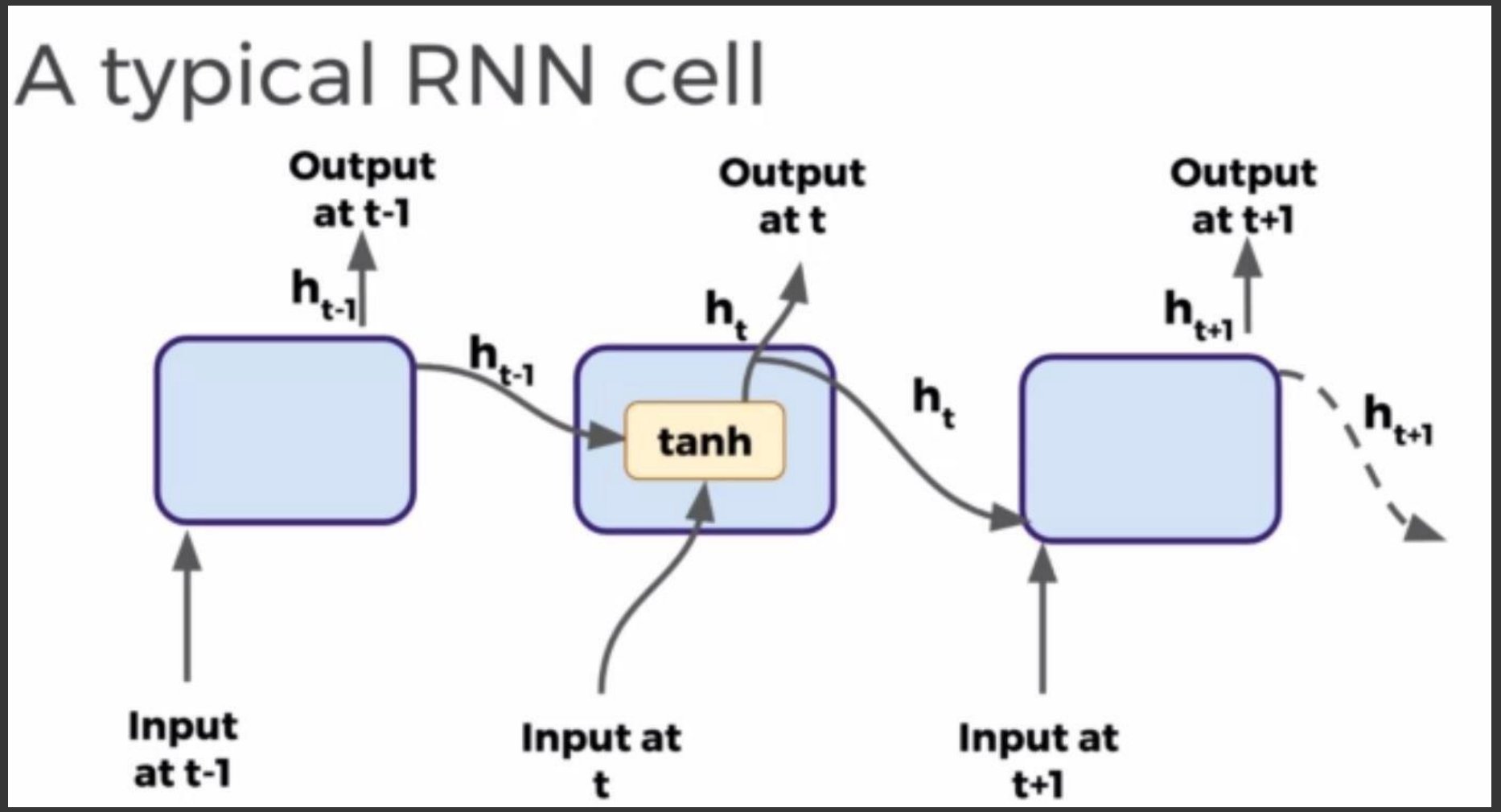
Input Layer

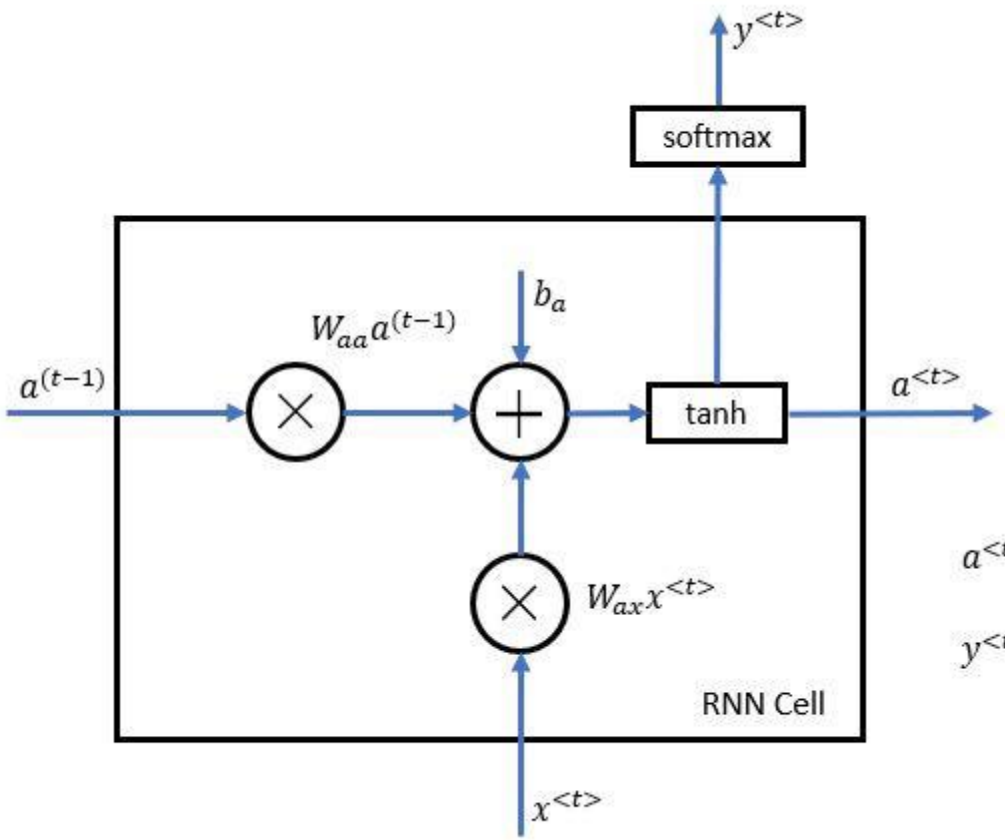
Hidden Layers

Output Layer

dataaspirant.com

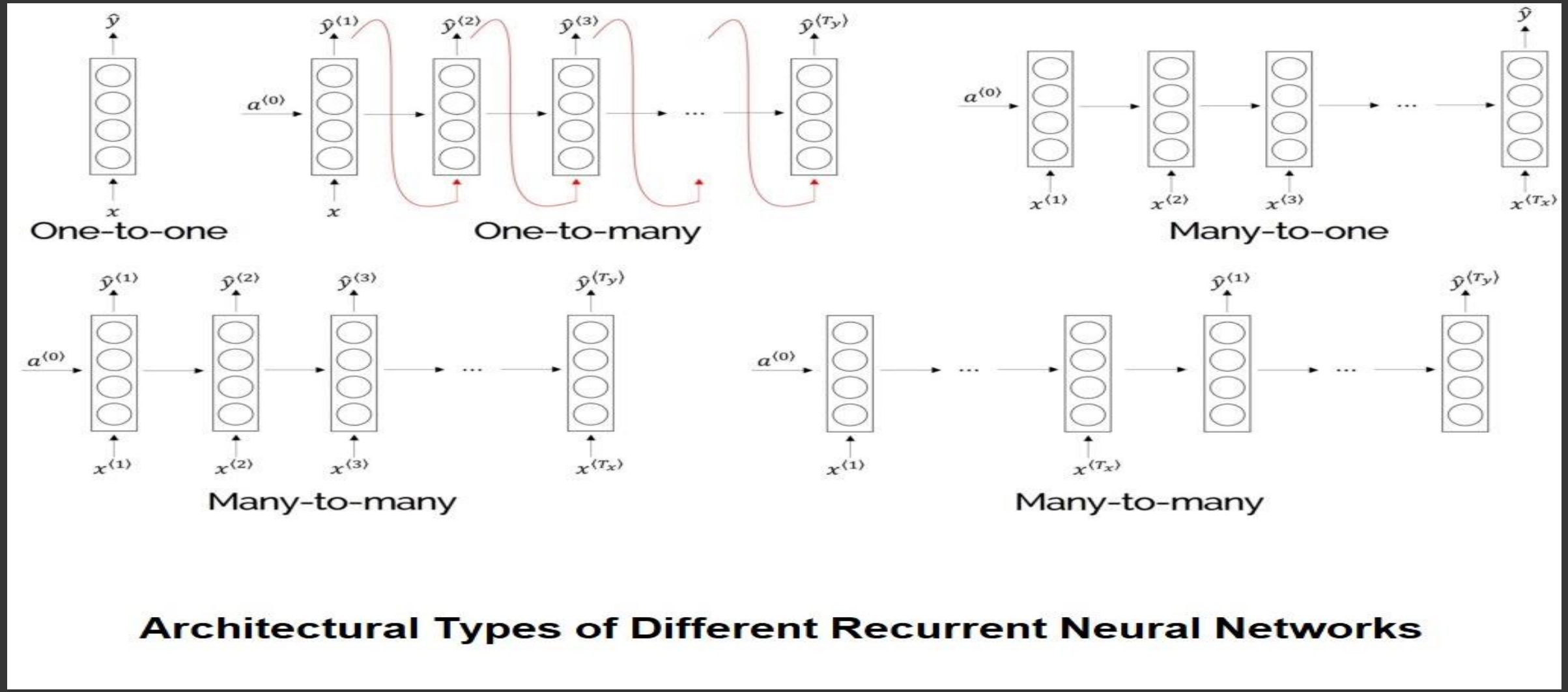
A typical RNN cell



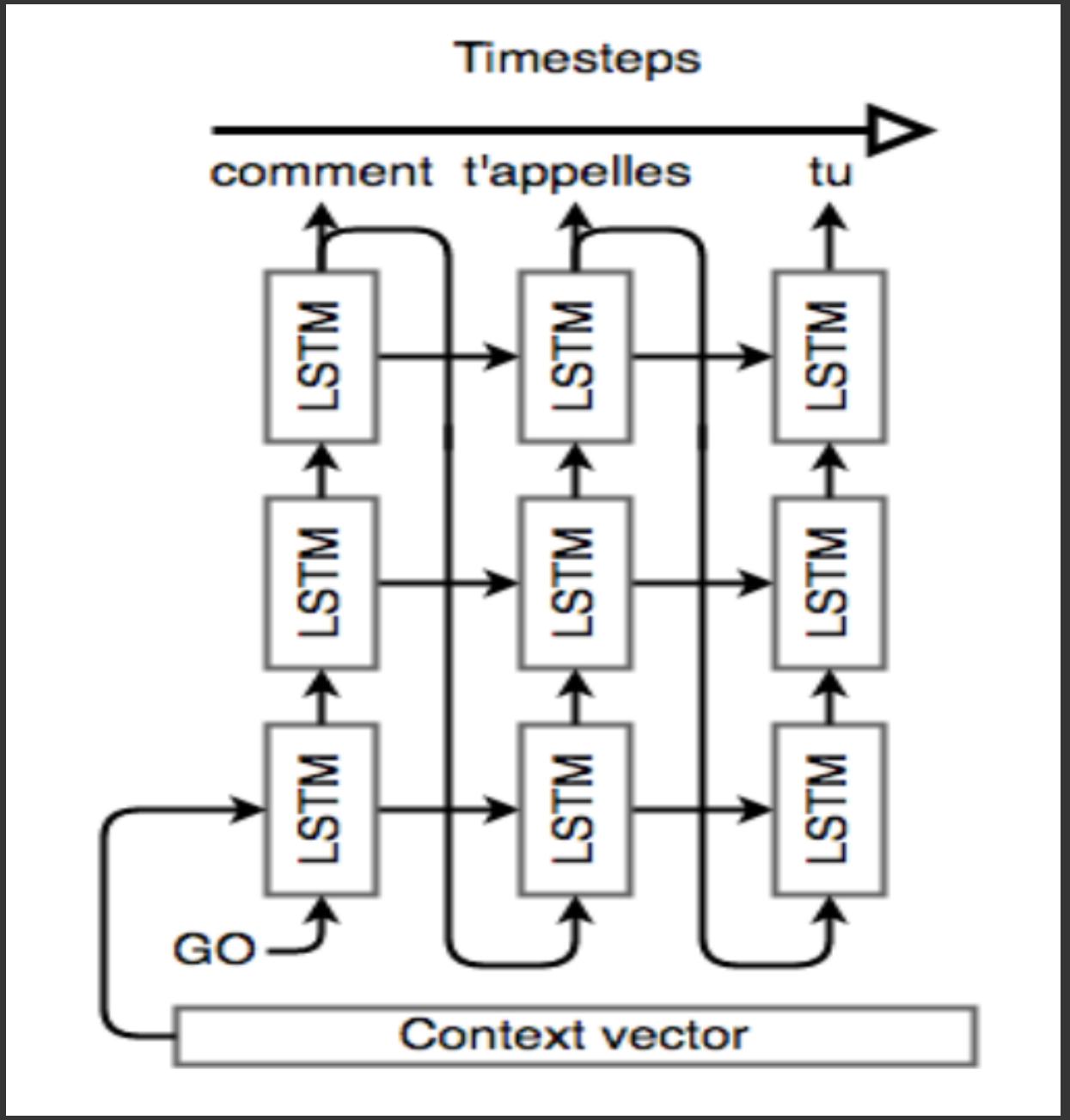


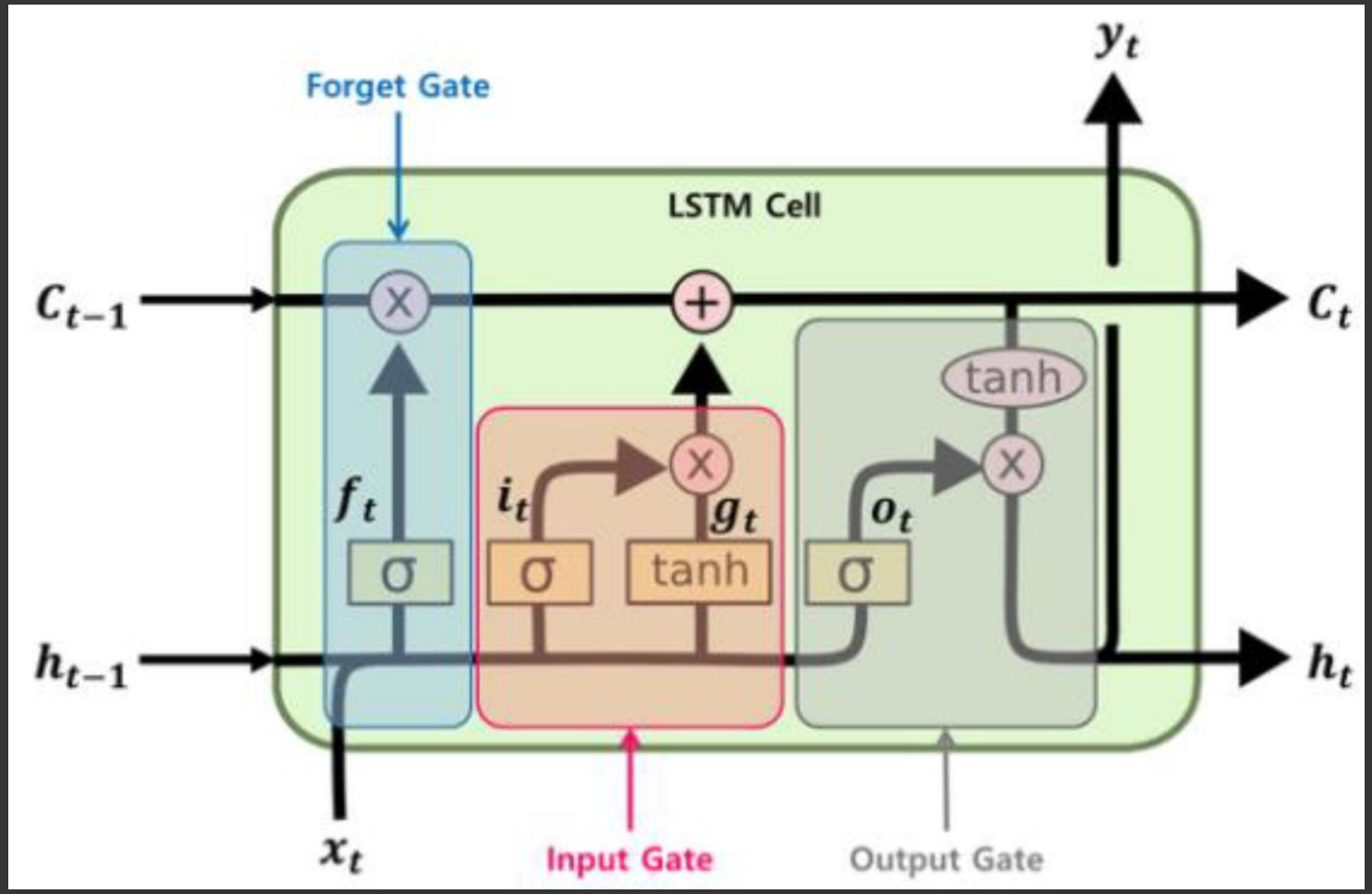
$$a^{<t>} = \tanh(W_{ax}x^{<t>} + W_{aa}a^{(t-1)} + b_a)$$

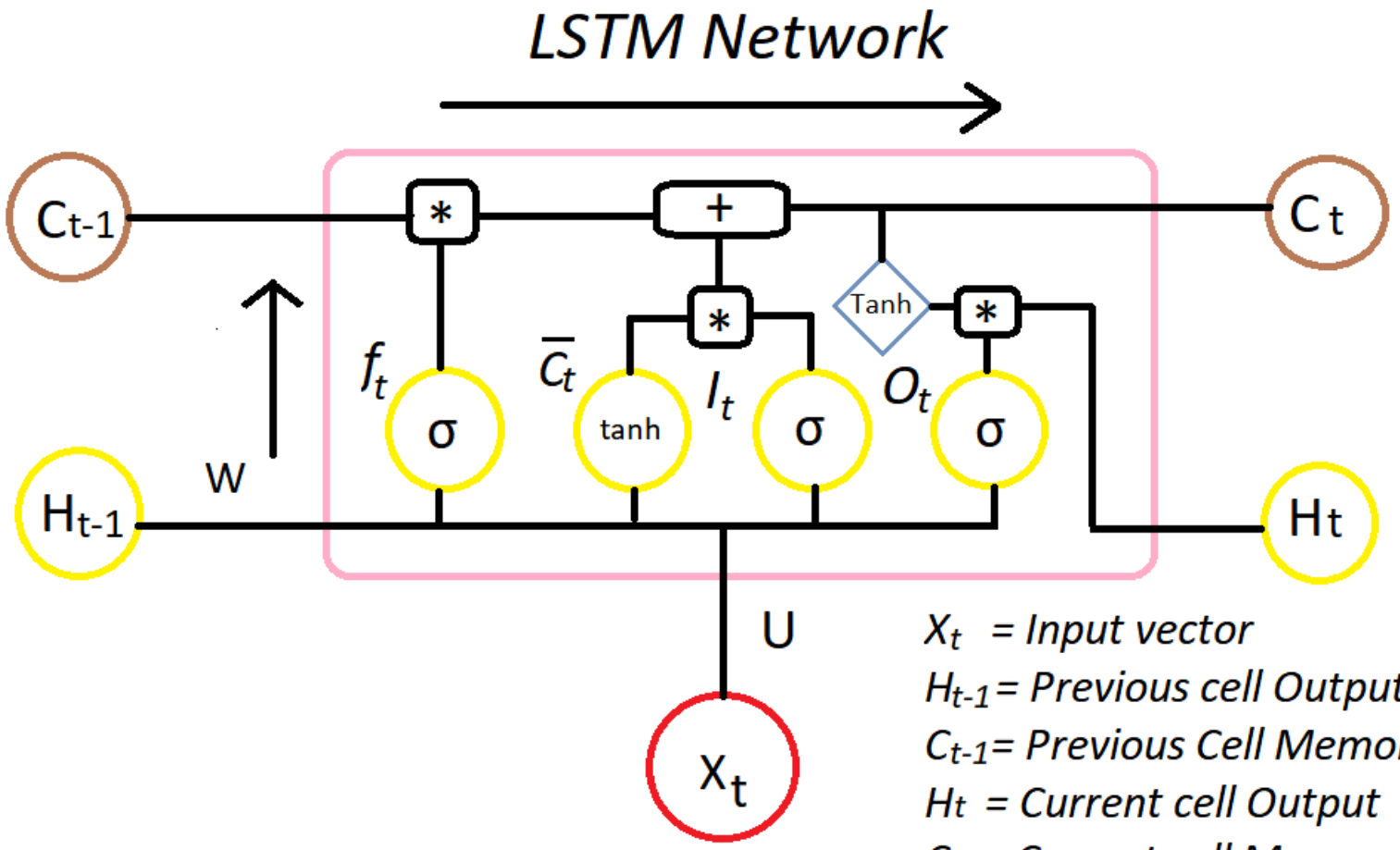
$$y^{<t>} = softmax(W_{ya}a^{<t>} + b_y)$$



Architectural Types of Different Recurrent Neural Networks


Timesteps
comment
t'appelles
tu
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
GO
Context vector


y_t
Forget Gate
LSTM Cell
C_{t-1}
C_t
tanh
f_t
i_t
g_t
o_t
σ
σ
tanh
σ
h_{t-1}
h_t
x_t
Input Gate
Output Gate



X_t = Input vector

H_{t-1} = Previous cell Output

C_{t-1} = Previous Cell Memory

H_t = Current cell Output

C_t = Current cell Memory

(*) = Element-wise multiplication

(+) = Element-wise addition

$$f_t = \sigma (X_t * U_f + H_{t-1} * W_f)$$
$$\bar{C}_t = \tanh (X_t * U_c + H_{t-1} * W_c)$$
$$I_t = \sigma (X_t * U_i + H_{t-1} * W_i)$$
$$O_t = \sigma (X_t * U_o + H_{t-1} * W_o)$$

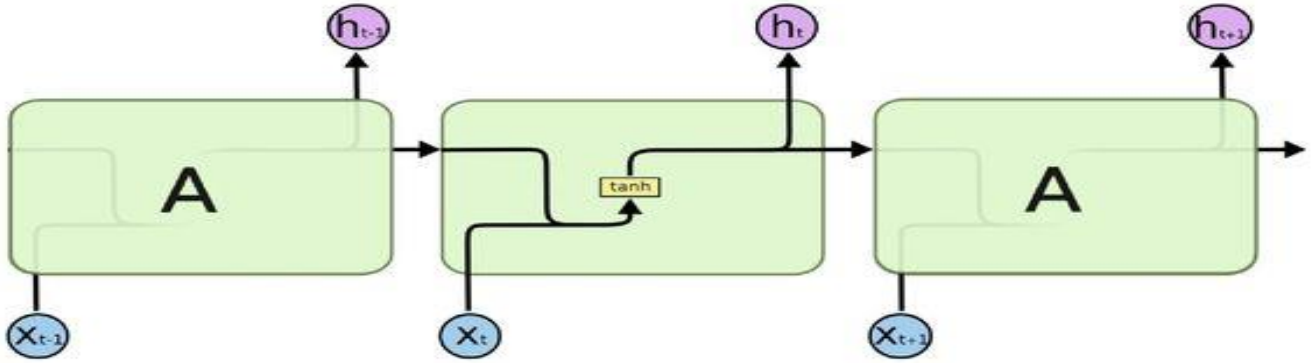
$$C_t = f_t * C_{t-1} + I_t * \bar{C}_t$$
$$H_t = O_t * \tanh (C_t)$$

W, U = weight vectors for forget gate (f), candidate (c) , i/p gate (I) and o/p gate (O)

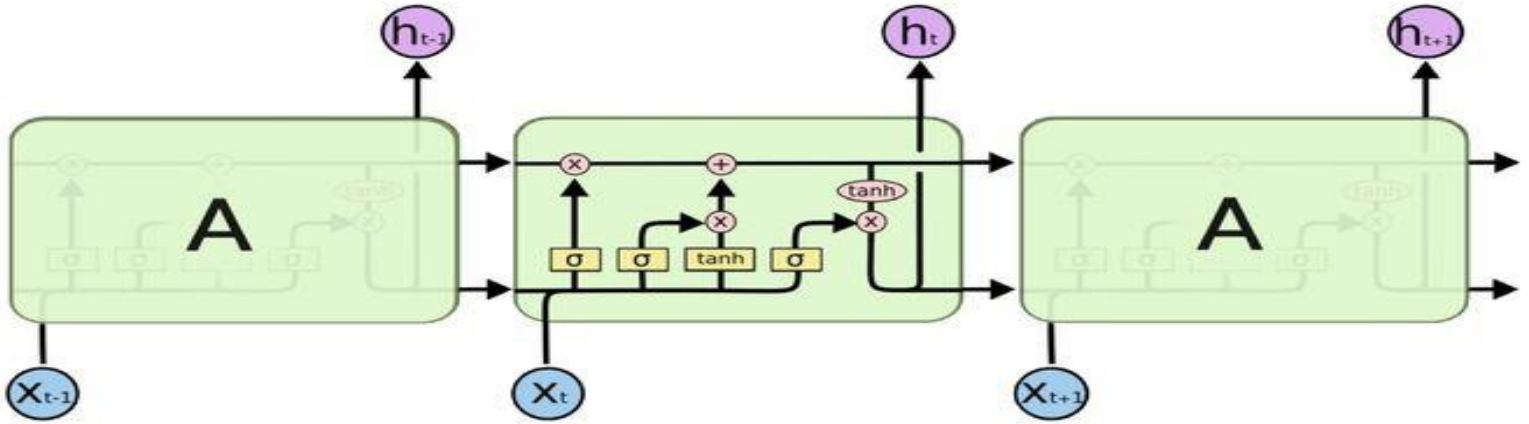
Note : These are different weights for different gates, for simpicity's sake, I mentioned W and U

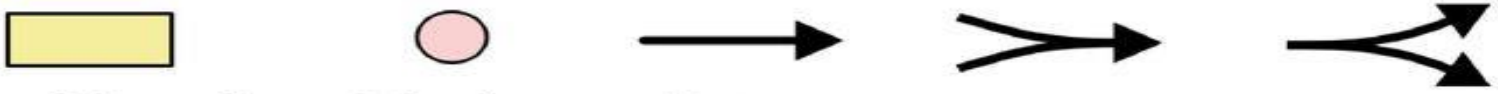
LSTM (Long short-term memory)

- Standard RNN
- Input concatenate with output then feed to input again



- LSTM
- The repeating structure is more complicated

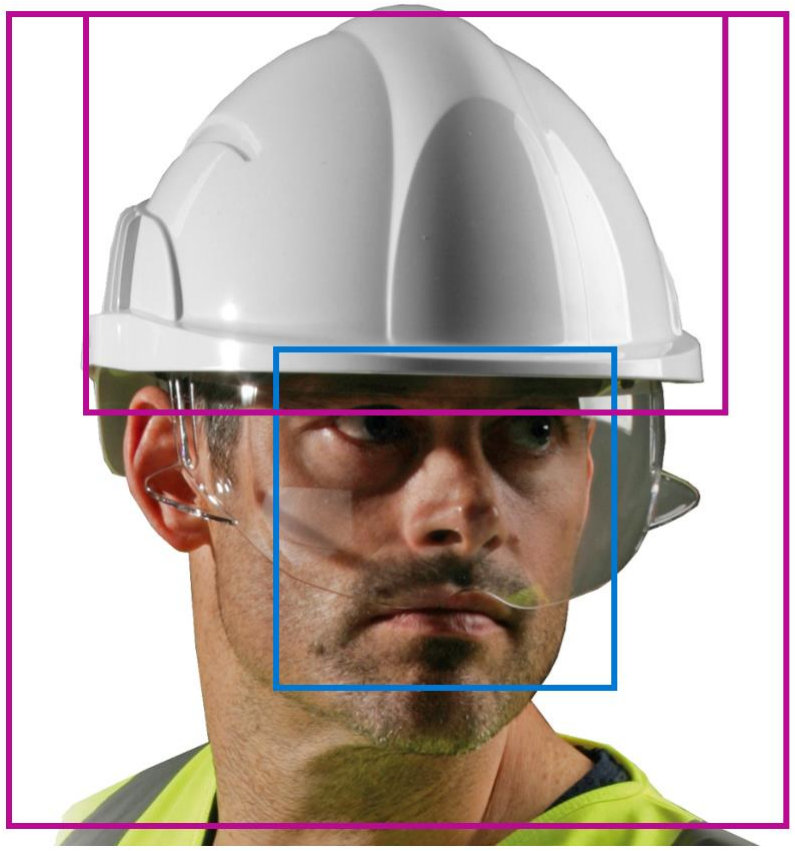




Adapting Computer Vision models with your own data

Transfer learning with Azure Cognitive Services Custom Vision

Cognitive Services Computer Vision



FEATURE NAME:	VALUE
Objects	[{ "rectangle": { "x": 138, "y": 27, "w": 746, "h": 471 }, "object": "headwear", "confidence": 0.616 }, { "rectangle": { "x": 52, "y": 33, "w": 910, "h": 951 }, "object": "person", "confidence": 0.802 }]
Tags	[{ "name": "man", "confidence": 0.999212 }, { "name": "headdress", "confidence": 0.99731946 }, { "name": "person", "confidence": 0.995057464 }, { "name": "clothing", "confidence": 0.991814733 }, { "name": "wearing", "confidence": 0.9827137 }, { "name": "hat", "confidence": 0.9691986 }, { "name": "helmet", "confidence": 0.9227209 }, { "name": "headgear", "confidence": 0.840476155 }, { "name": "personal protective equipment", "confidence": 0.8358513 }, { "name": "looking", "confidence": 0.832229853 }, { "name": "hard hat", "confidence": 0.8004248 }, { "name": "human face", "confidence": 0.785058737 }, { "name": "green", "confidence":

Transfer Learning: Training

F1
F2
F3
F4
F5
F6
F7
F8

hammer
1

hard hat
0

Transfer Learning: Training

F1
F2
F3
F4
F5
F6
F7
F8

hammer
0

hard hat
1

Transfer Learning: Trained Model

F1
F2
F3
F4
F5
F6
F7
F8

hammer
0.10

hard hat
0.87

Thank you!

Q & A