Track 7

Inteligencia Artificial Aplicada

4. DNN – Regression (Hands-On with CoLab)

Prof. Marcelo José Rovai
rovai@unifei.edu.br

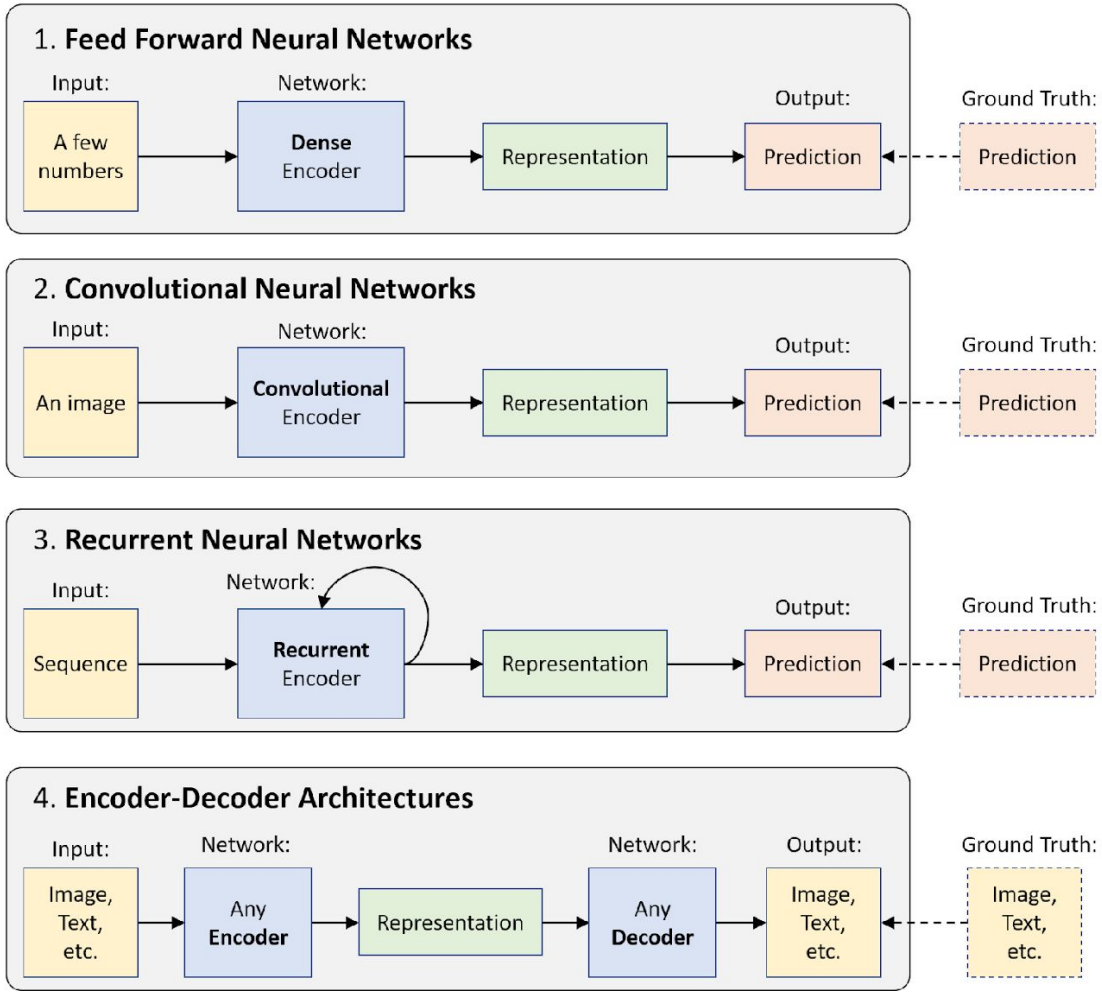
UNIFEI - Universidade Federal de Itajubá, Brazil

Machine Learning

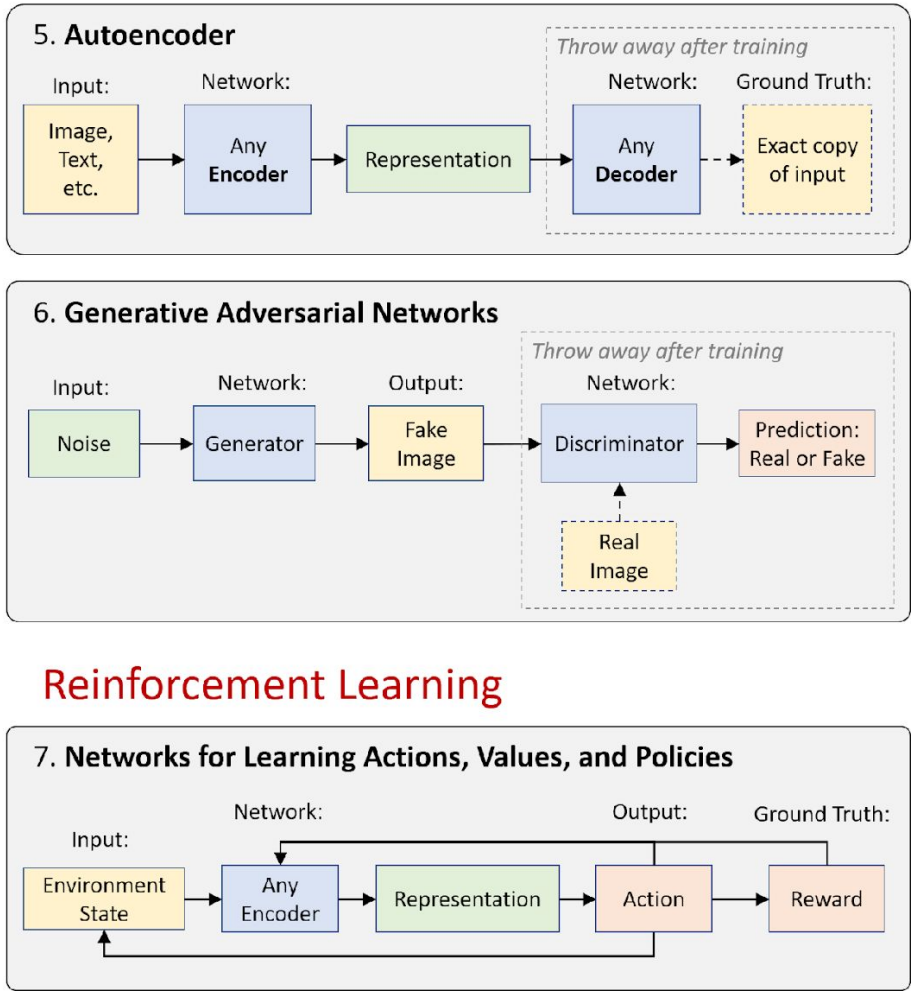
Models

Machine Learning Types and Architectures

Supervised Learning



Unsupervised Learning



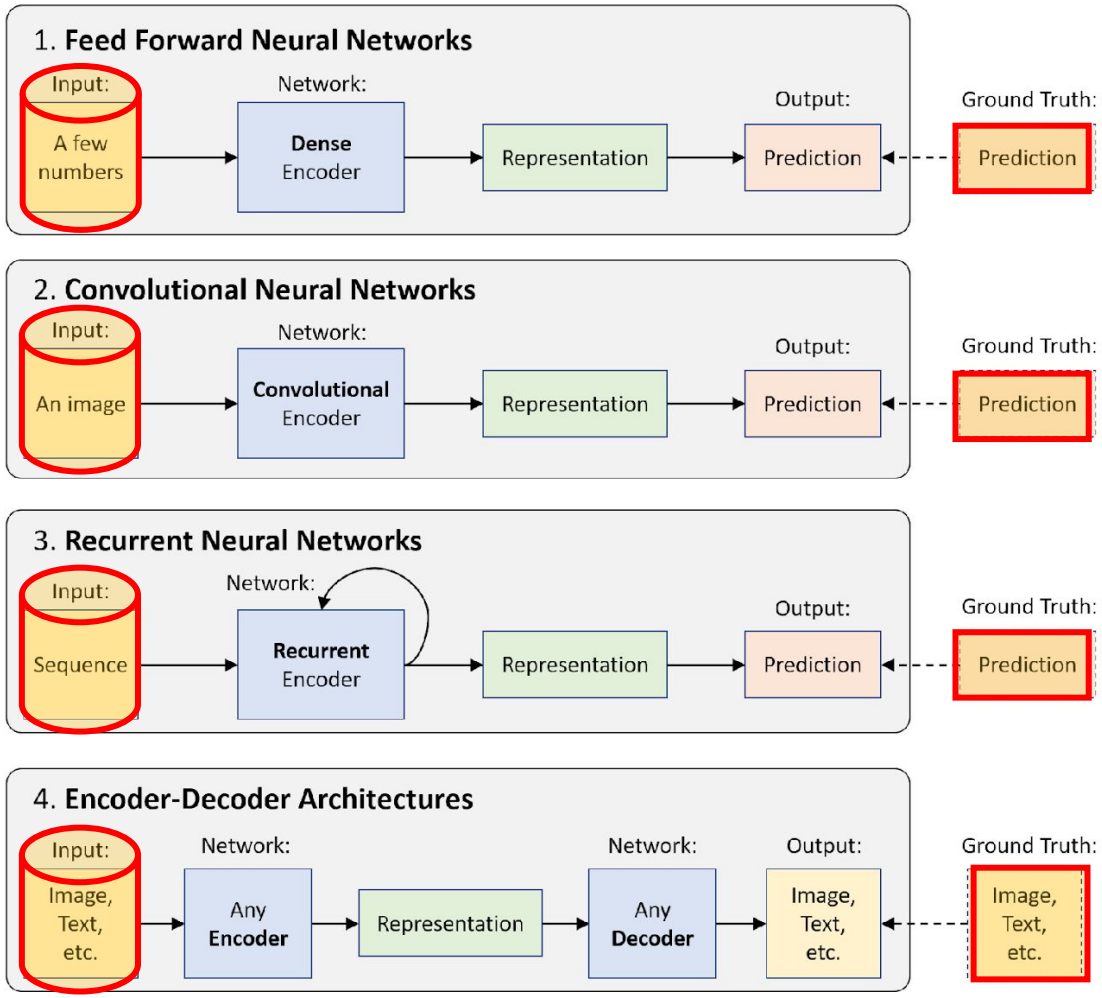
Reinforcement Learning

[Deep Learning Basics: An introductory lecture for MIT course 6.S094 by Prof. Lex Fridman]

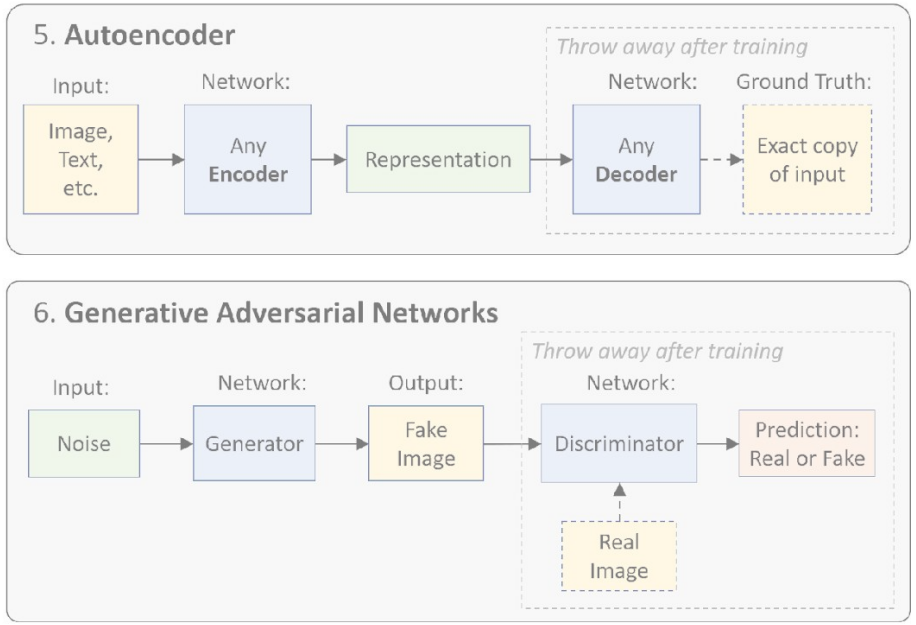
Machine Learning

Supervised Learning

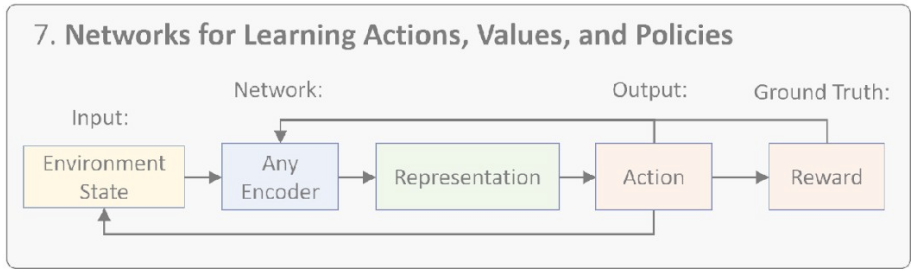
Training



Unsupervised Learning

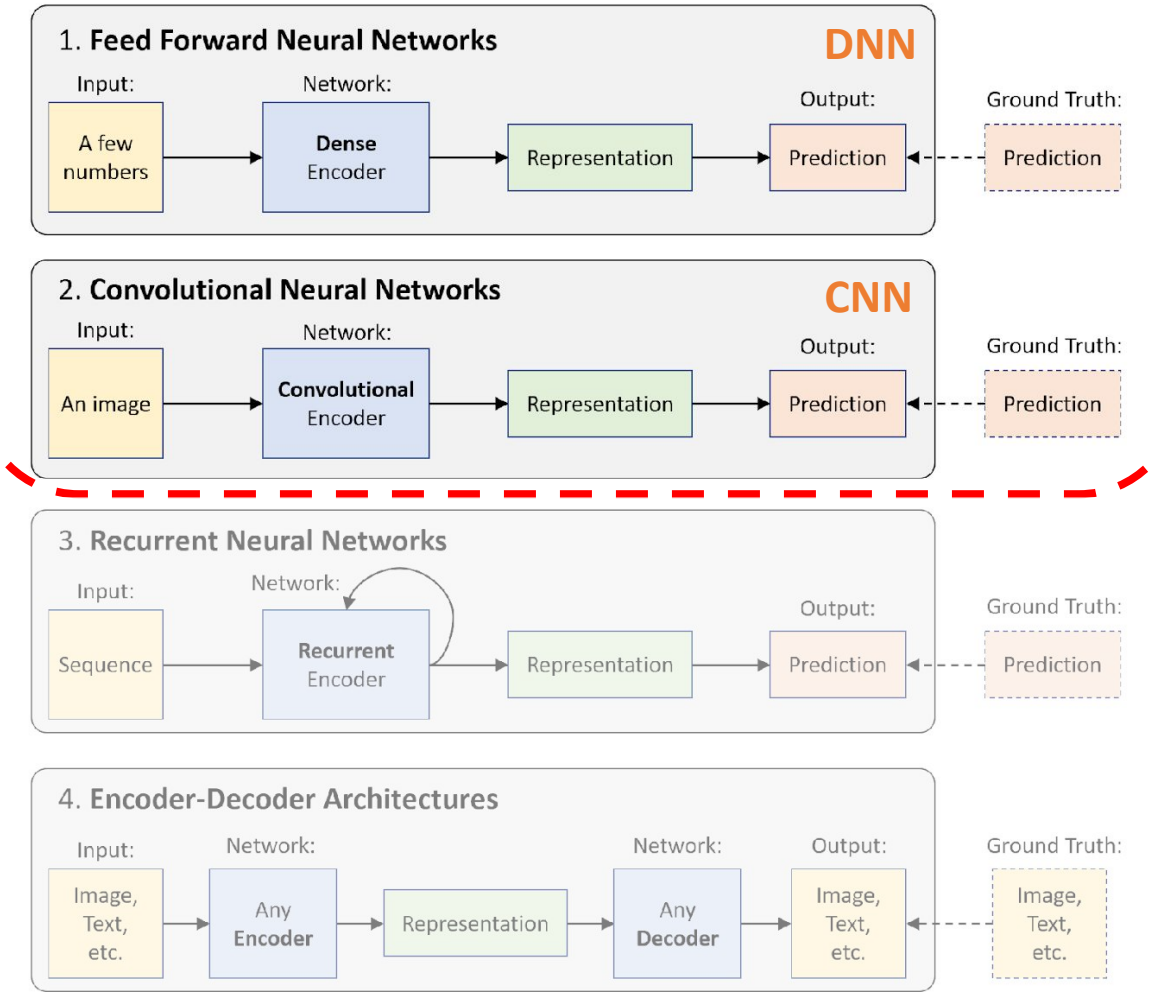


Reinforcement Learning

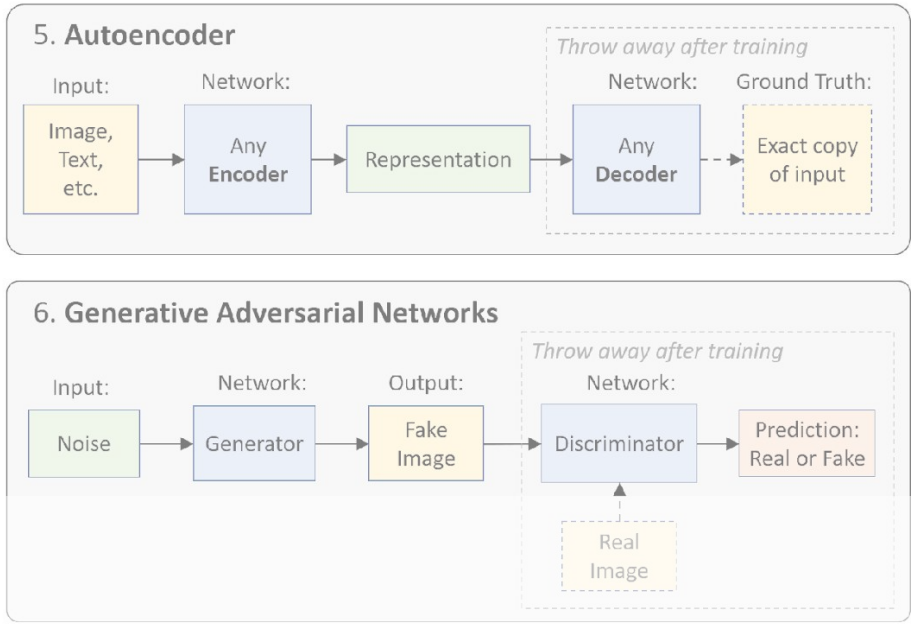


Machine Learning

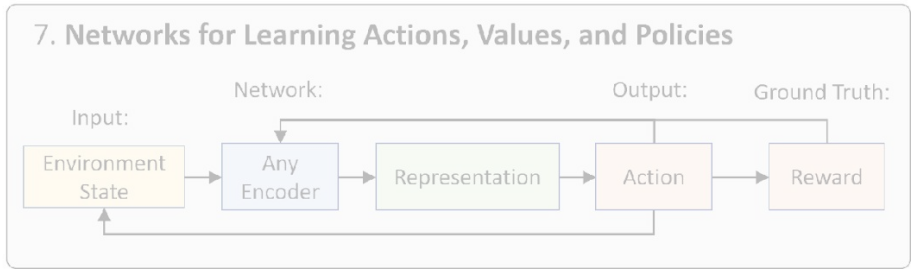
Supervised Learning



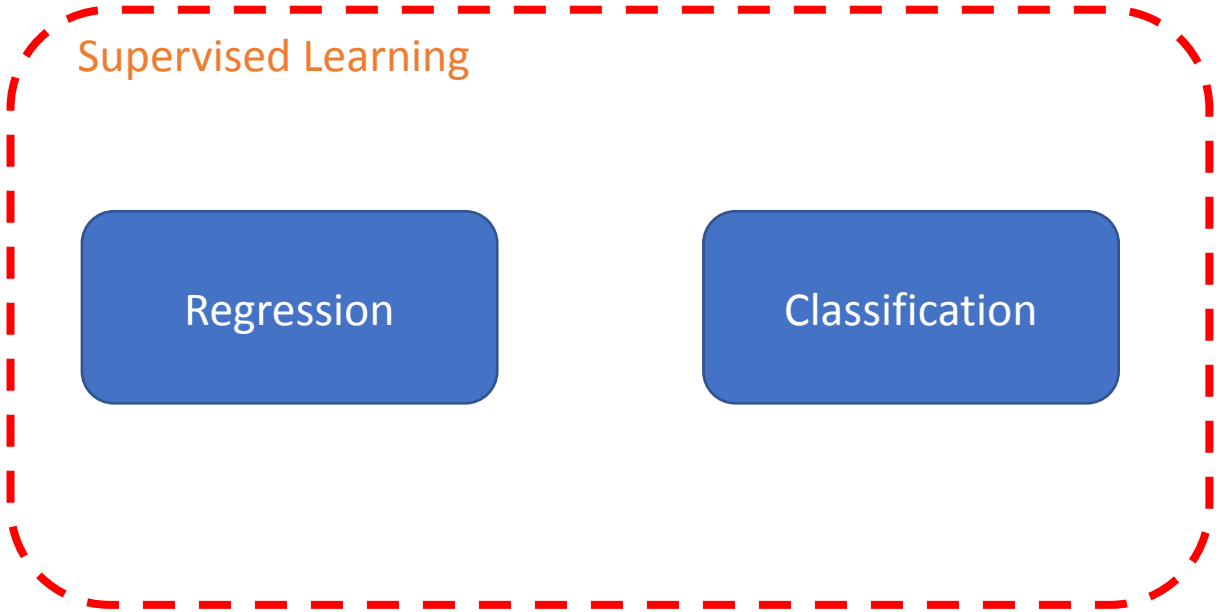
Unsupervised Learning



Reinforcement Learning



Tiny Machine Learning

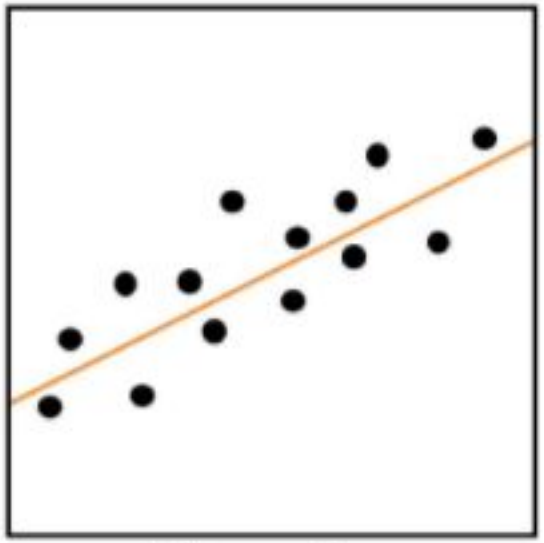


Tiny Machine Learning

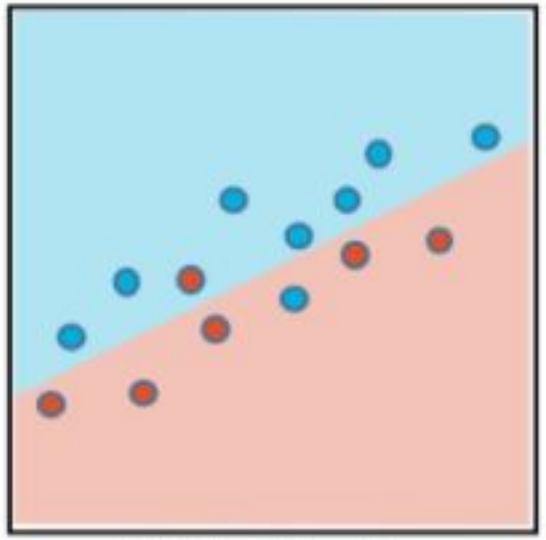
Supervised Learning

Regression

Classification



a) Regression

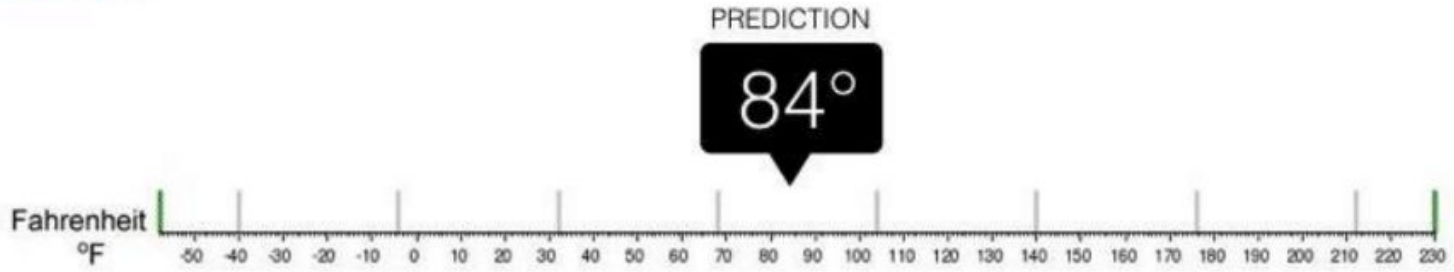


b) Classification

Regression

Regression

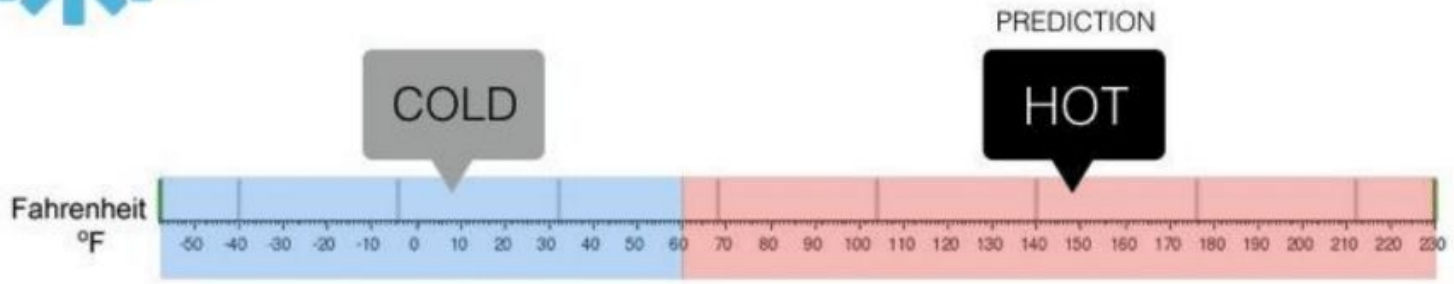
What is the temperature going to be tomorrow?



Classification

Classification

Will it be Cold or Hot tomorrow?



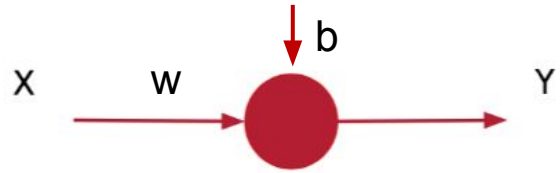
Machine Learning

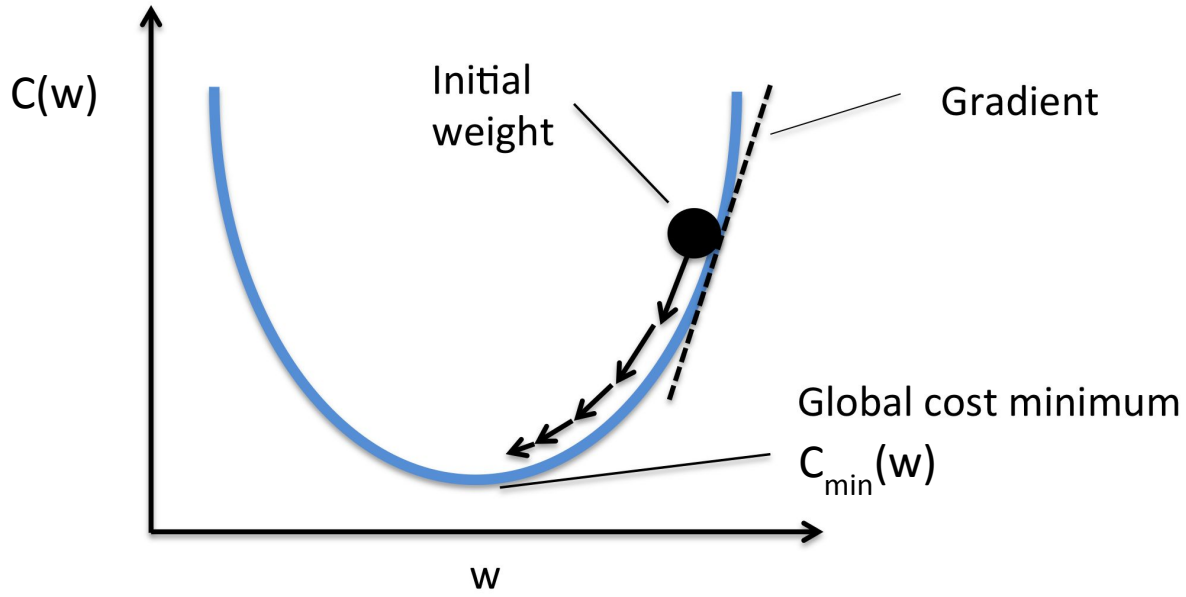
Supervised models - Regression

X ➡ -1, 0, 1, 2, 3, 4

Y ➡ -3, -1, 1, 3, 5, 7

X	Y
-1	-3
0	-1
1	1
2	3
3	5
4	7

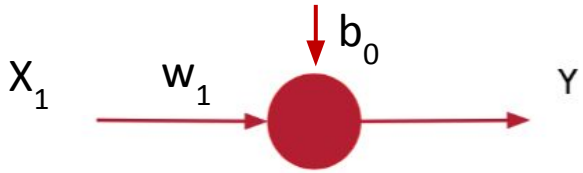


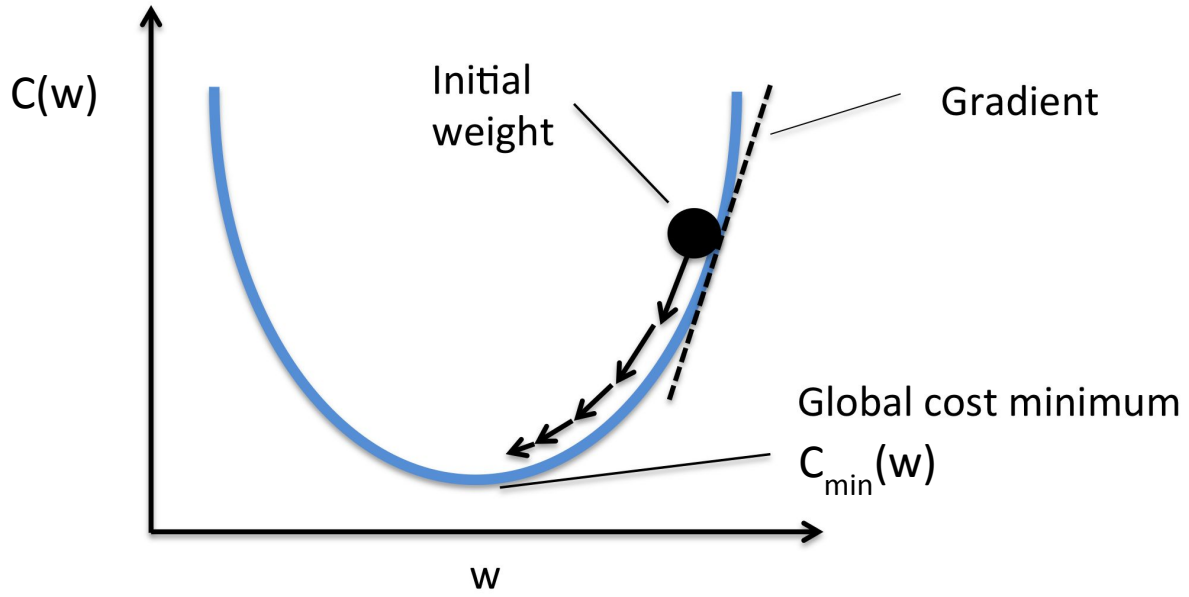


$$Y = w * X + b$$

Cost Function

X_1	Y
-1	-3
0	-1
1	1
2	3
3	5
4	7

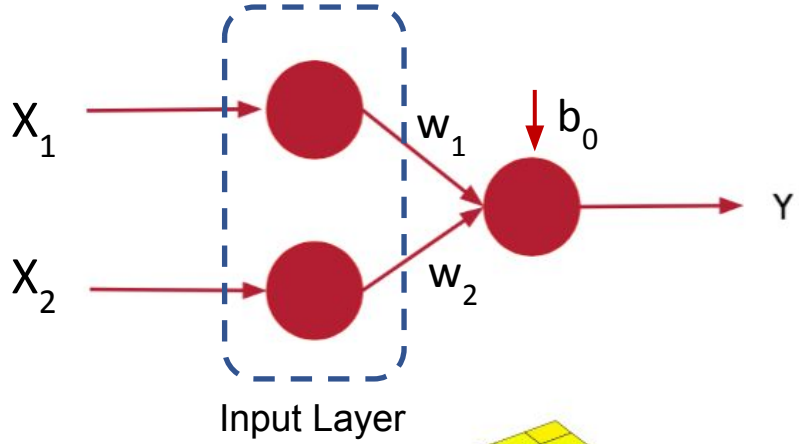


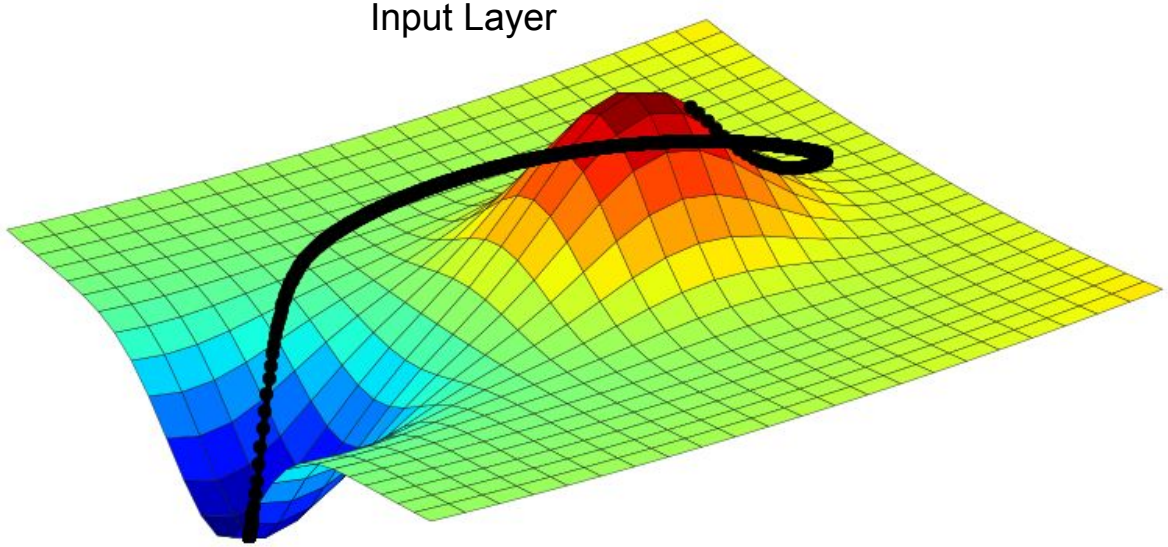


$$Y = w_1 * X_1 + b_0$$

Cost Function

X_1	X_2	Y
-1	-8	-8
0	1	0
1	3	7
2	7	1
3	0	2
4	2	3

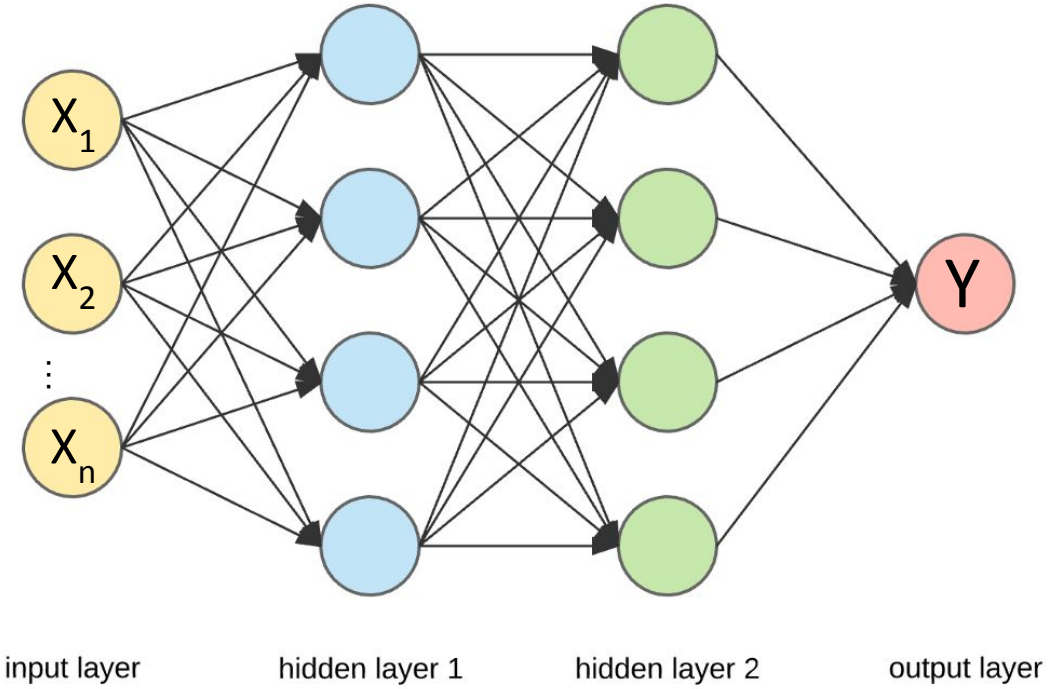




$$Y = w_1 * X_1 + w_2 * X_2 + b_0$$

Cost Function

X_1	X_2	…	X_n	Y
-1	-8		-81	-8
0	1		10	0
1	3		3	7
2	7		7	1
3	0		0	2
4	2		-7	3



$$Y = w_1 * X_1 + w_2 * X_2 + \ldots + w_n * X_n + b_0 + b_1 + \ldots + b_n$$

Regression using DNN with TF2

Code Time!

TF_Boston_Housing_Regression.ipynb

Machine Learning

Workflow

Machine Learning Workflow

Collect Data

```
data = tf.keras.datasets.boston_housing

(x_train, y_train), (x_test, y_test) = data.load_data()
```


17

Machine Learning Workflow


```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(x_train)

x_train_norm = scaler.transform(x_train)
x_test_norm = scaler.transform(x_test)
```


18

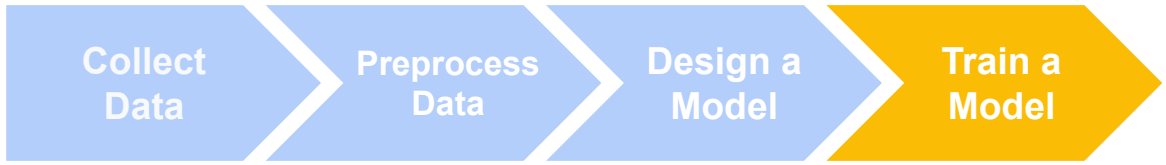
Machine Learning Workflow


```
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(20,
                          activation='relu',
                          input_shape = [13]),
    tf.keras.layers.Dense(1)
    ])
```

```
model.compile(
    optimizer='adam',
    loss='mse',
    metrics=['mae']
    )
```


19

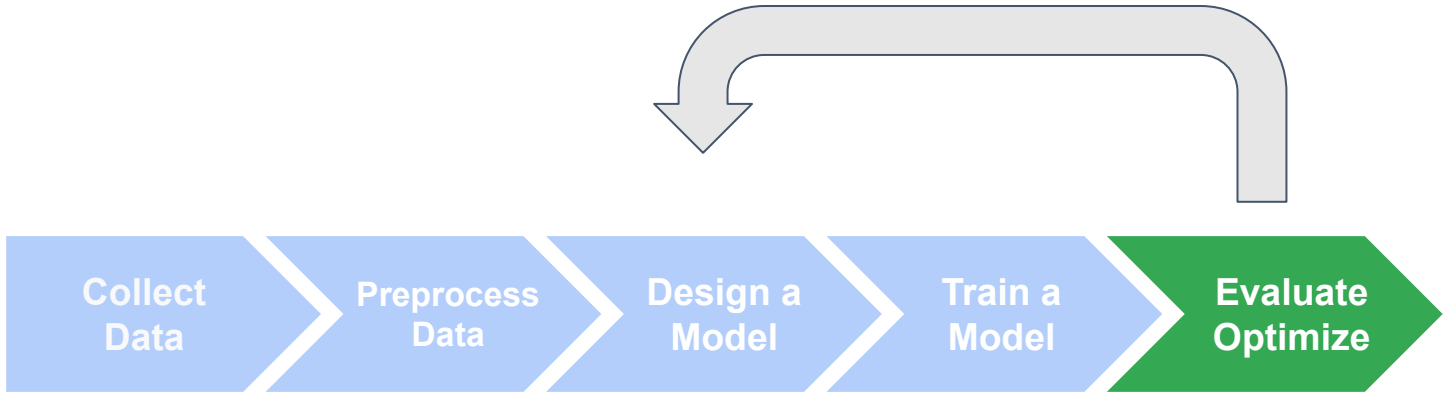
Machine Learning Workflow

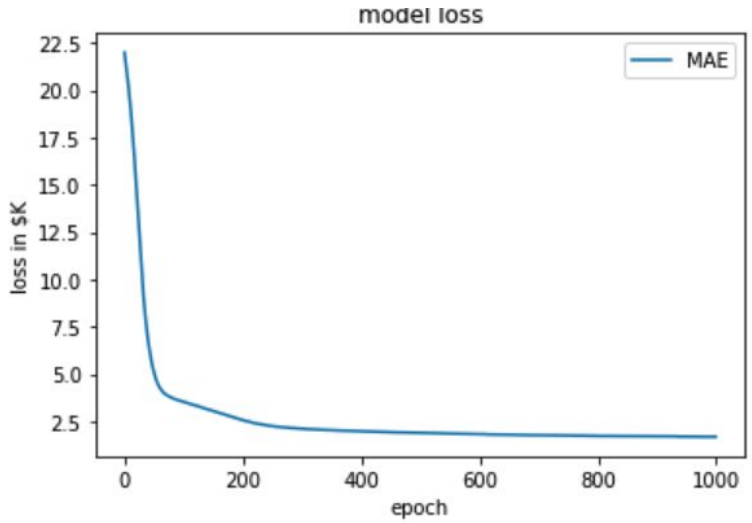



```
history = model.fit(
    x_train_norm,
    y_train,
    epochs=1000,
    verbose=0
    )
```


20

Machine Learning Workflow






```
train_eval = model.evaluate(x_train_norm, y_train)
print ("Training data MSE: {:.2}".format(train_eval[1]))
```

```
tuner.search(
    x_train_norm, y_train,
    epochs=500,
    validation_data=(x_test_norm, y_test))
```

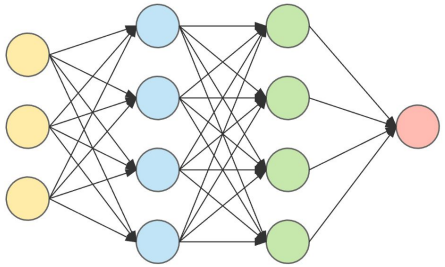

21

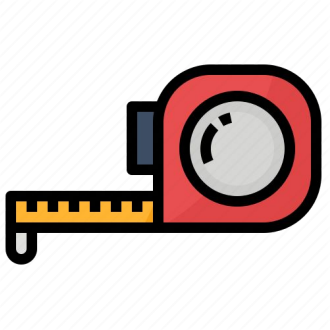
Machine Learning Workflow

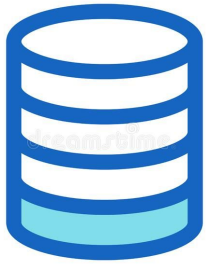

```
xt = np.array([1.1, 0., 9., 0., 0.6, 7., 92., 3.8 , 4., 300., 21., 200, 19.5])
xt = np.reshape(xt, (1, 13))
xt_norm = scaler.transform(xt)
yt = model.predict(xt_norm)
```


22

Machine Learning Workflow







23

Thanks

UNIFEI