

IESTI01 – TinyML

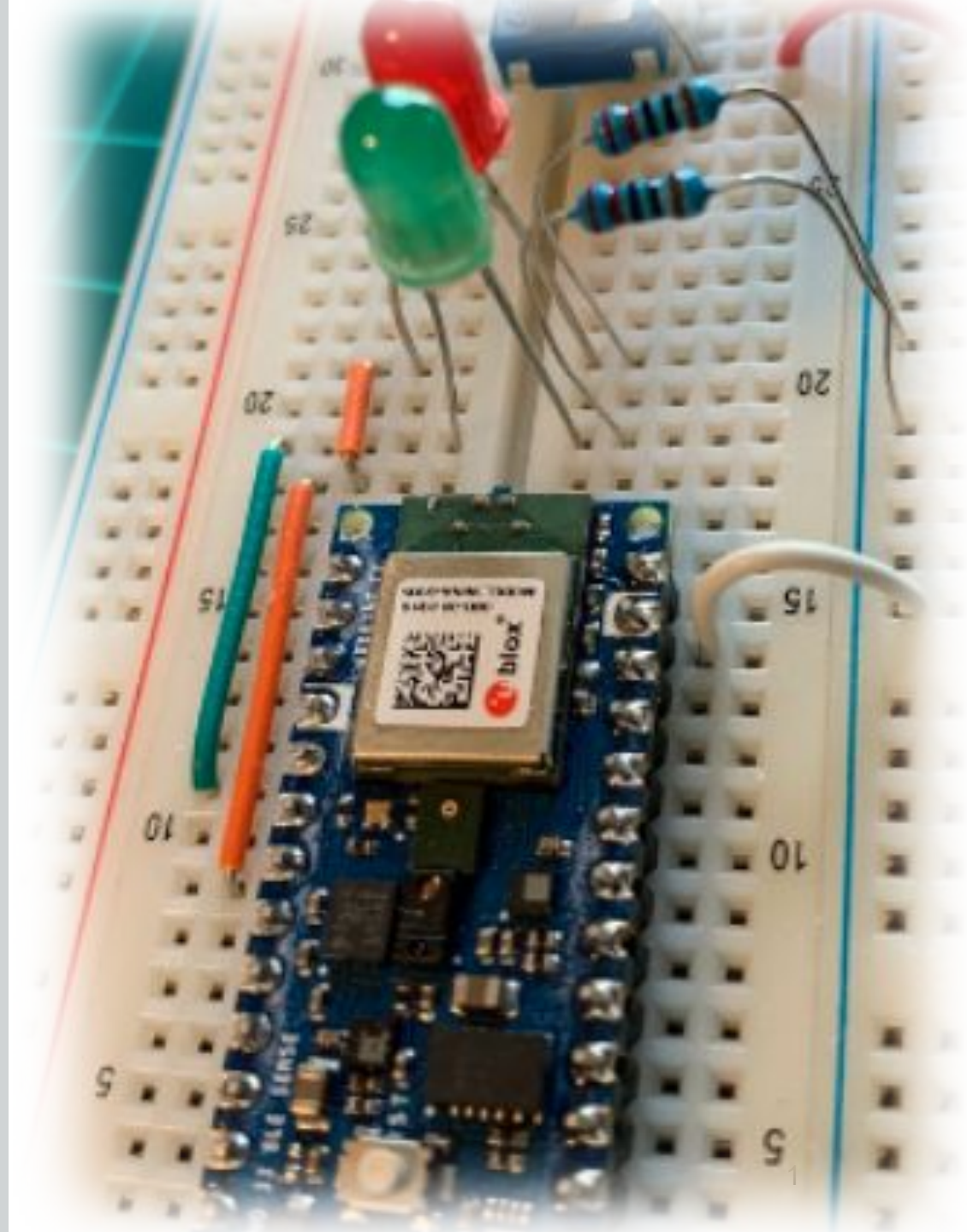
Embedded Machine Learning

7. The Building Blocks of Deep Learning – Part A a. Regression



Prof. Marcelo Rovai

UNIFEI

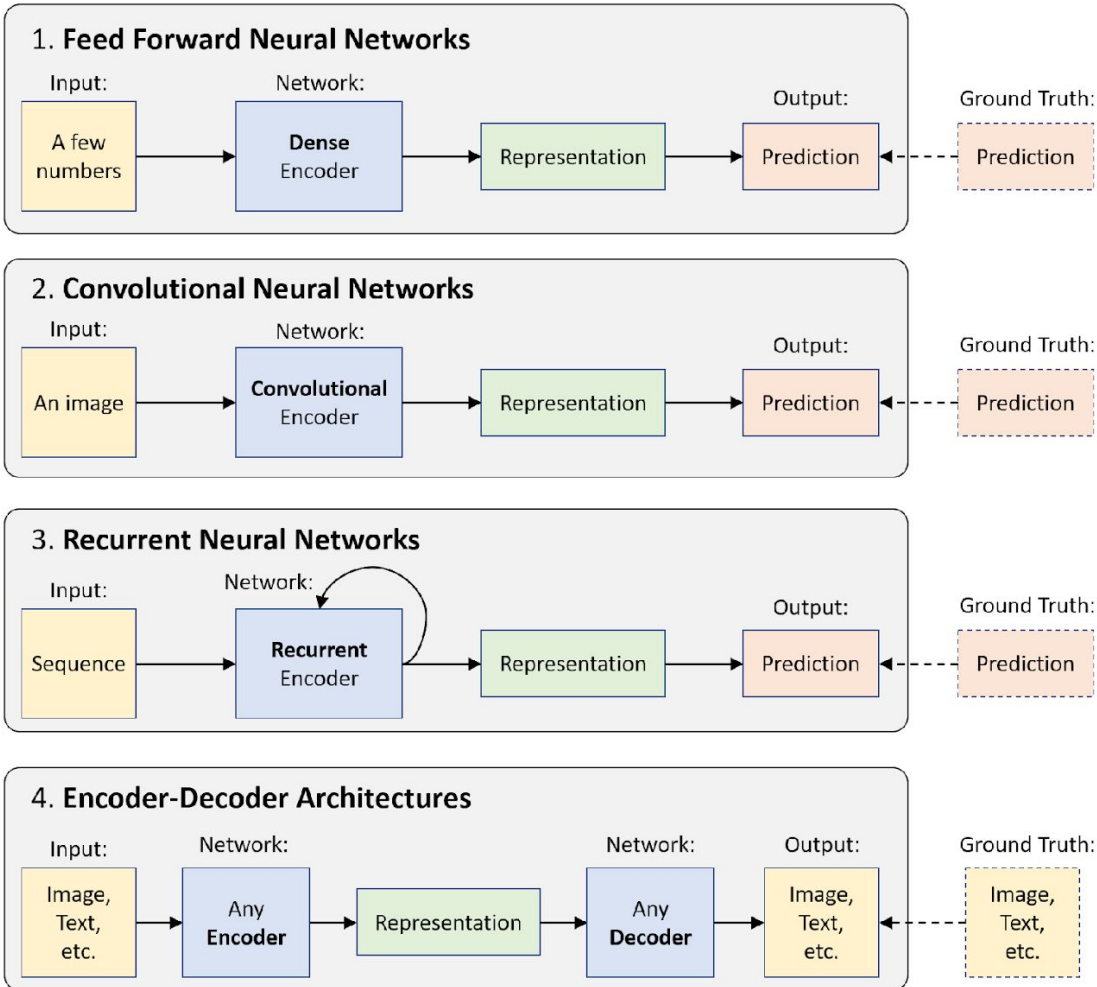


Machine Learning

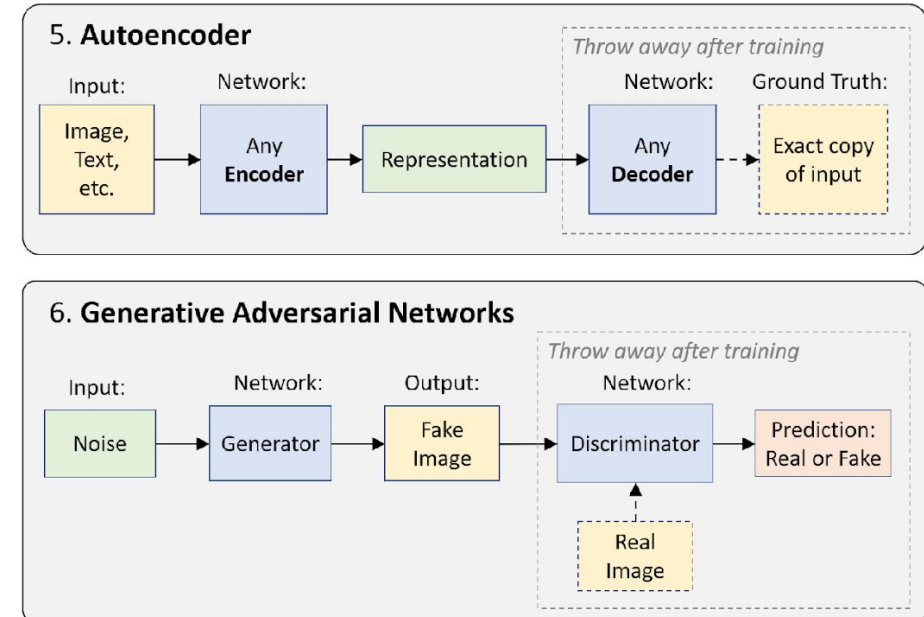
Models

Machine Learning Types and Architectures

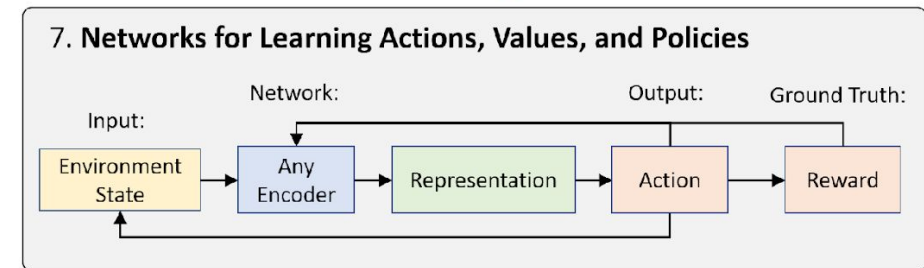
Supervised Learning



Unsupervised Learning



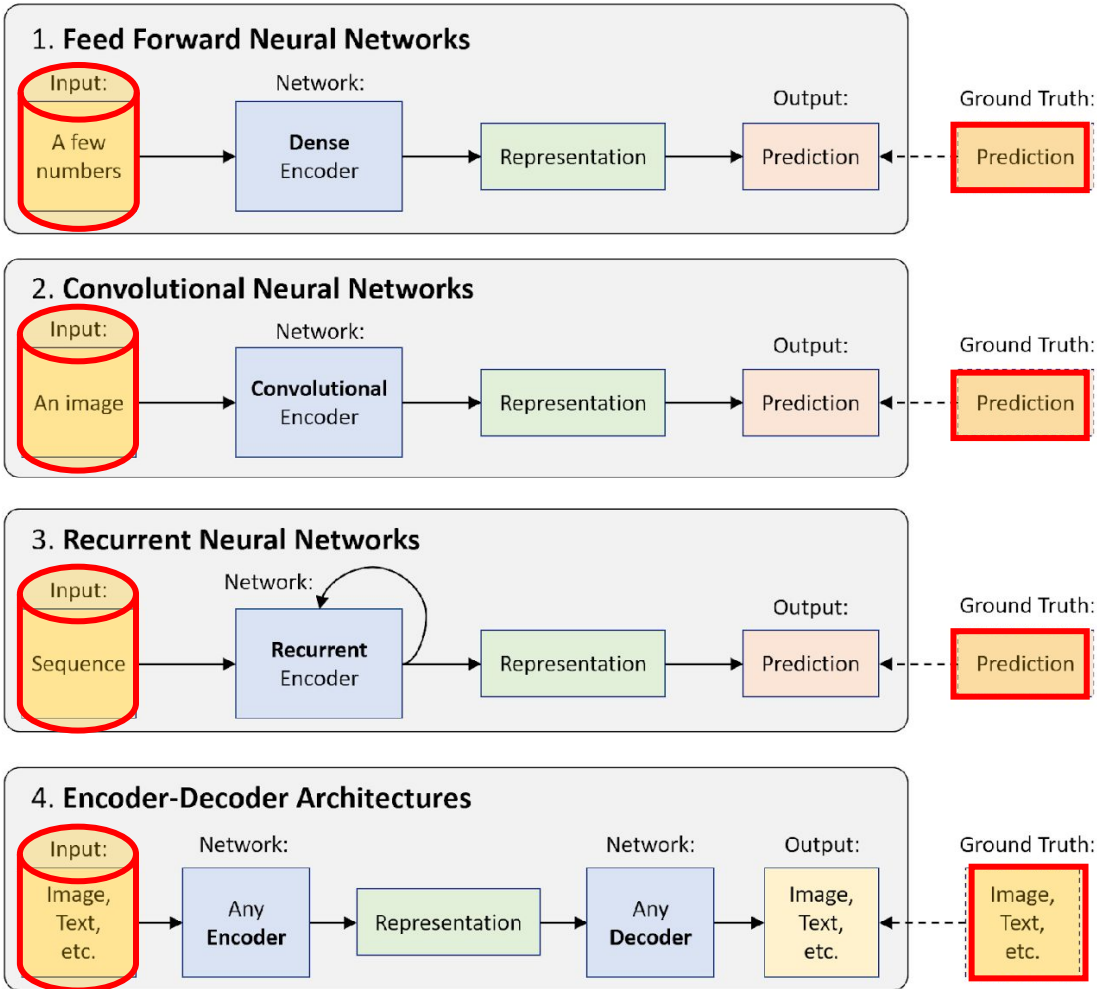
Reinforcement Learning



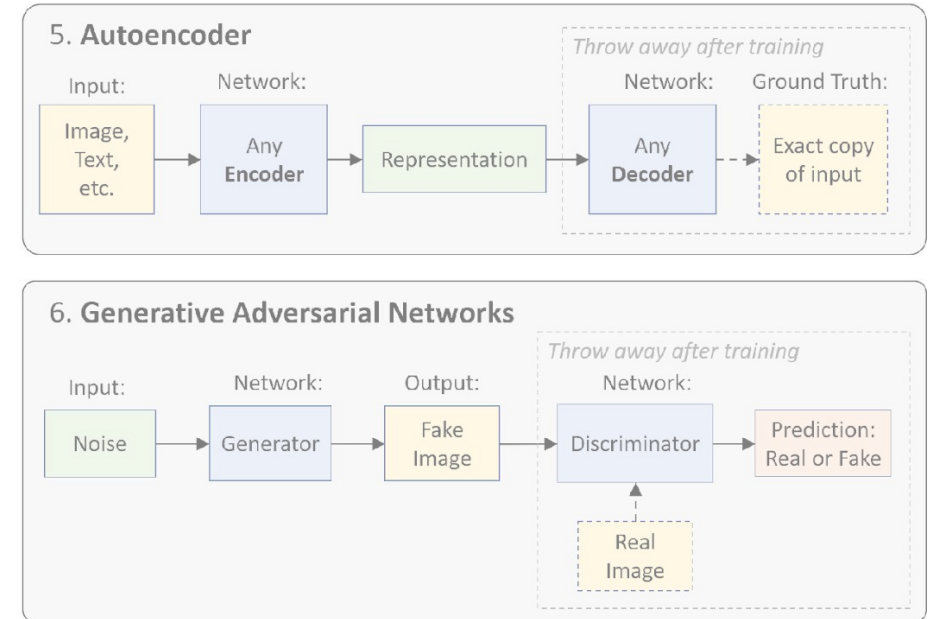
Machine Learning

Supervised Learning

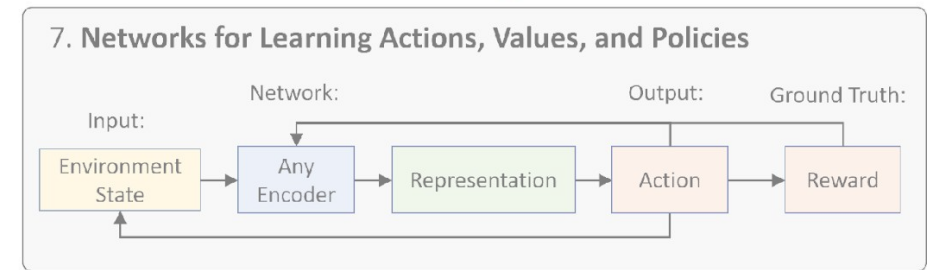
Training



Unsupervised Learning

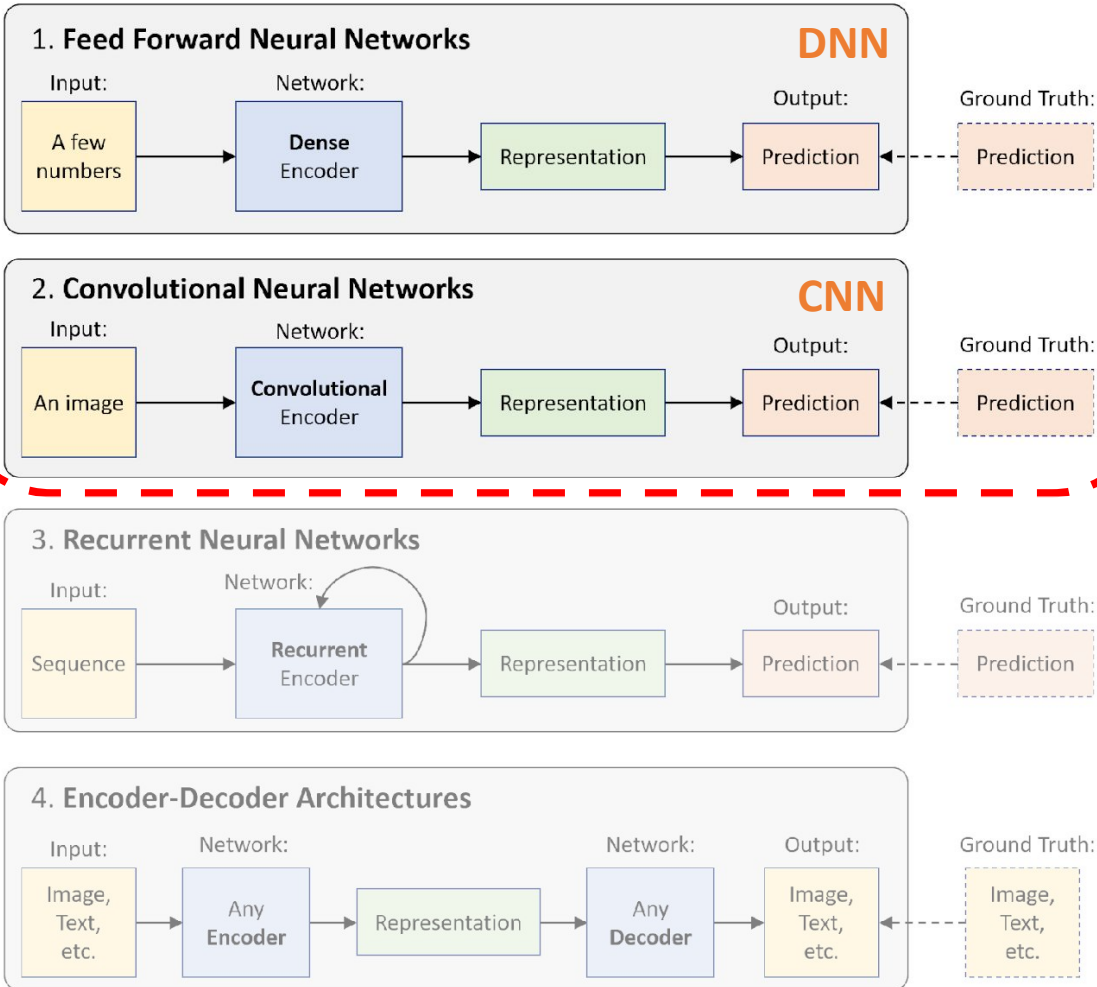


Reinforcement Learning

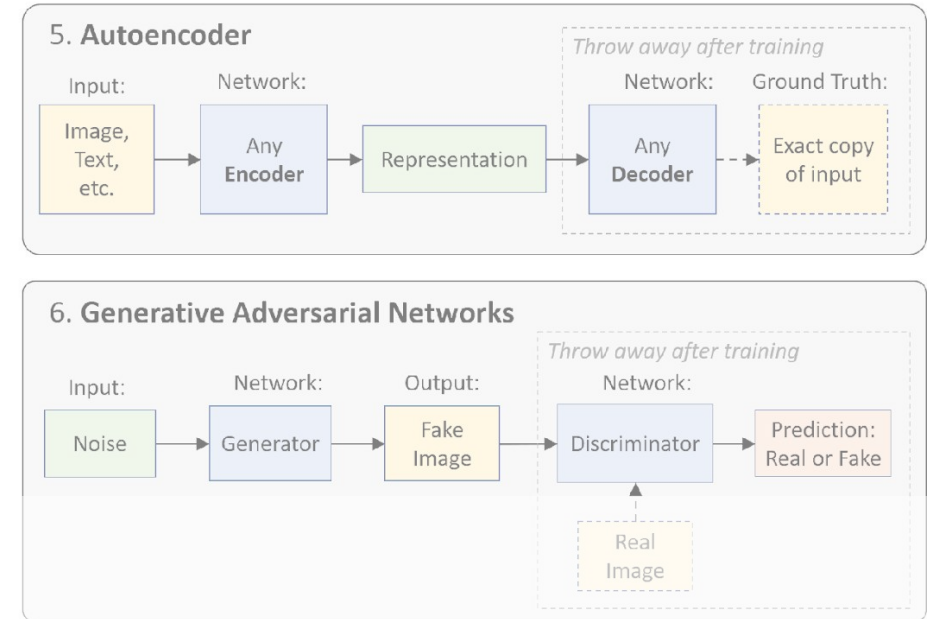


Machine Learning

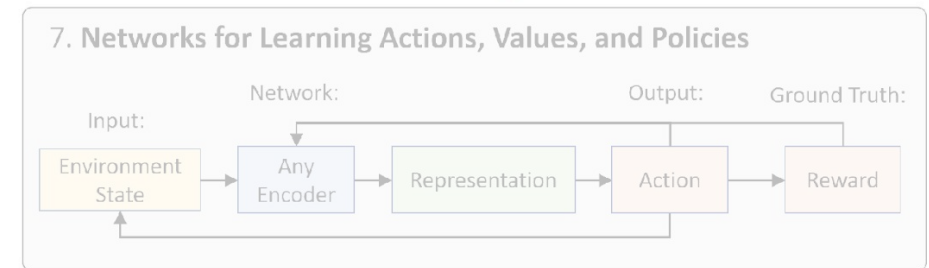
Supervised Learning



Unsupervised Learning



Reinforcement Learning



Tiny Machine Learning

Supervised Learning

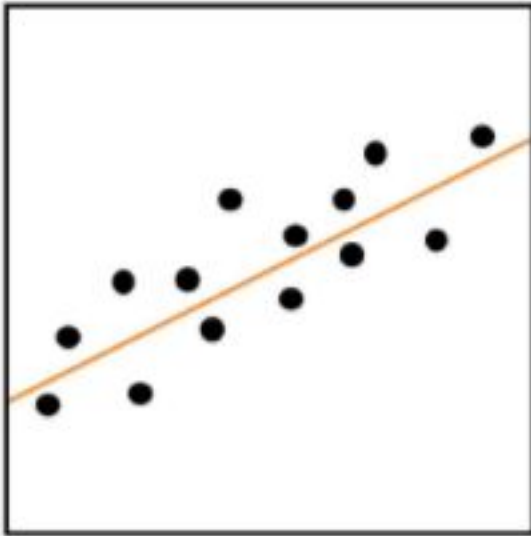
Regression

Classification

Tiny Machine Learning

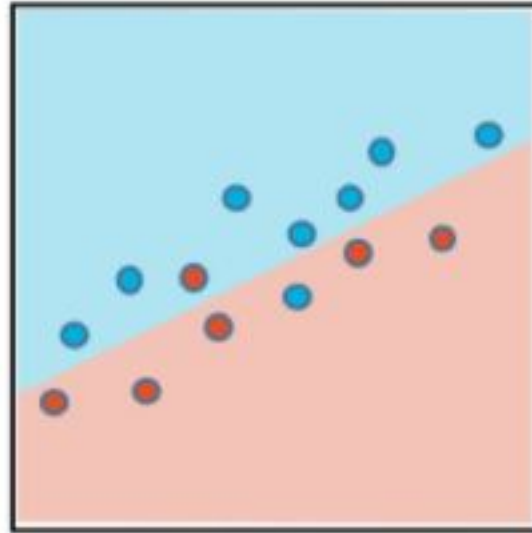
Supervised Learning

Regression



a) Regression

Classification



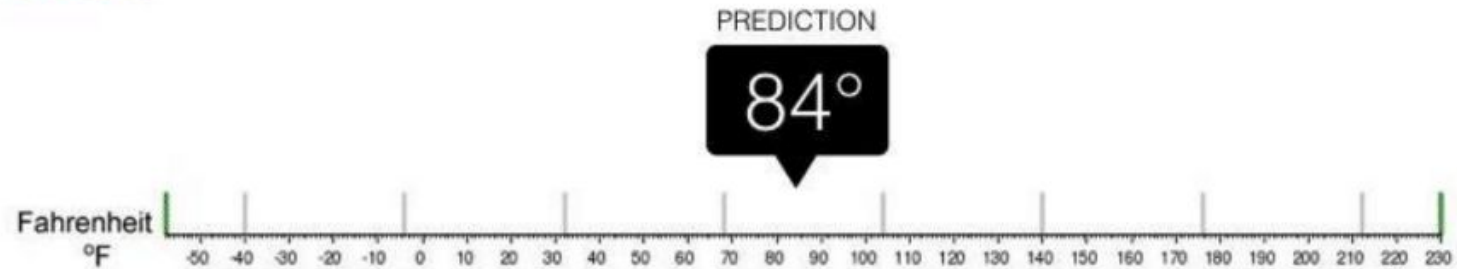
b) Classification

Regression



Regression

What is the temperature going to be tomorrow?

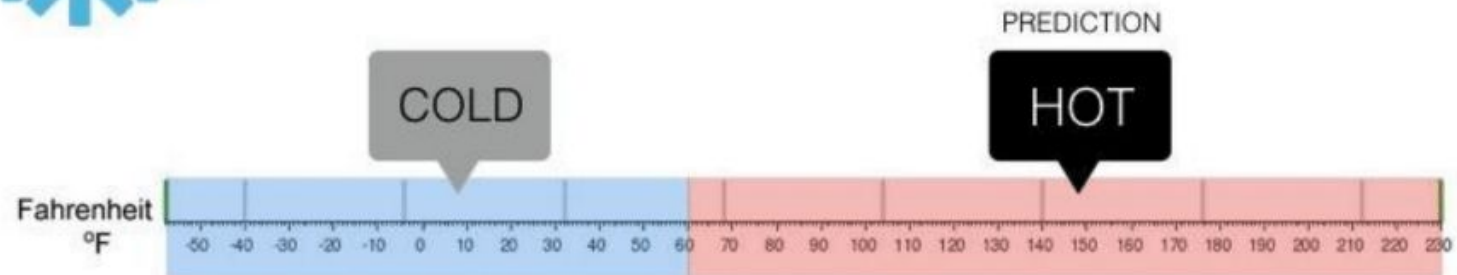


Classification



Classification

Will it be Cold or Hot tomorrow?



Machine Learning

Supervised models - Regression

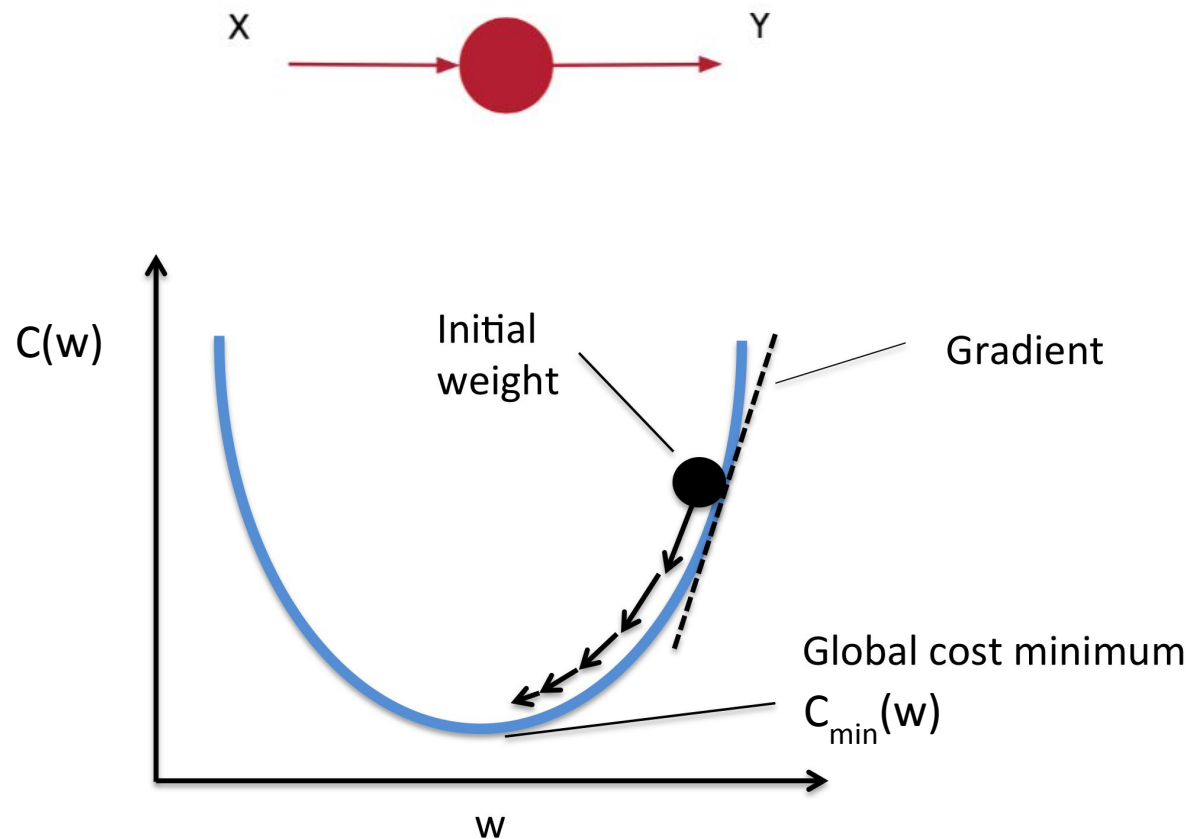
$X \rightarrow -1, 0, 1, 2, 3, 4$

$Y \rightarrow -3, -1, 1, 3, 5, 7$



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

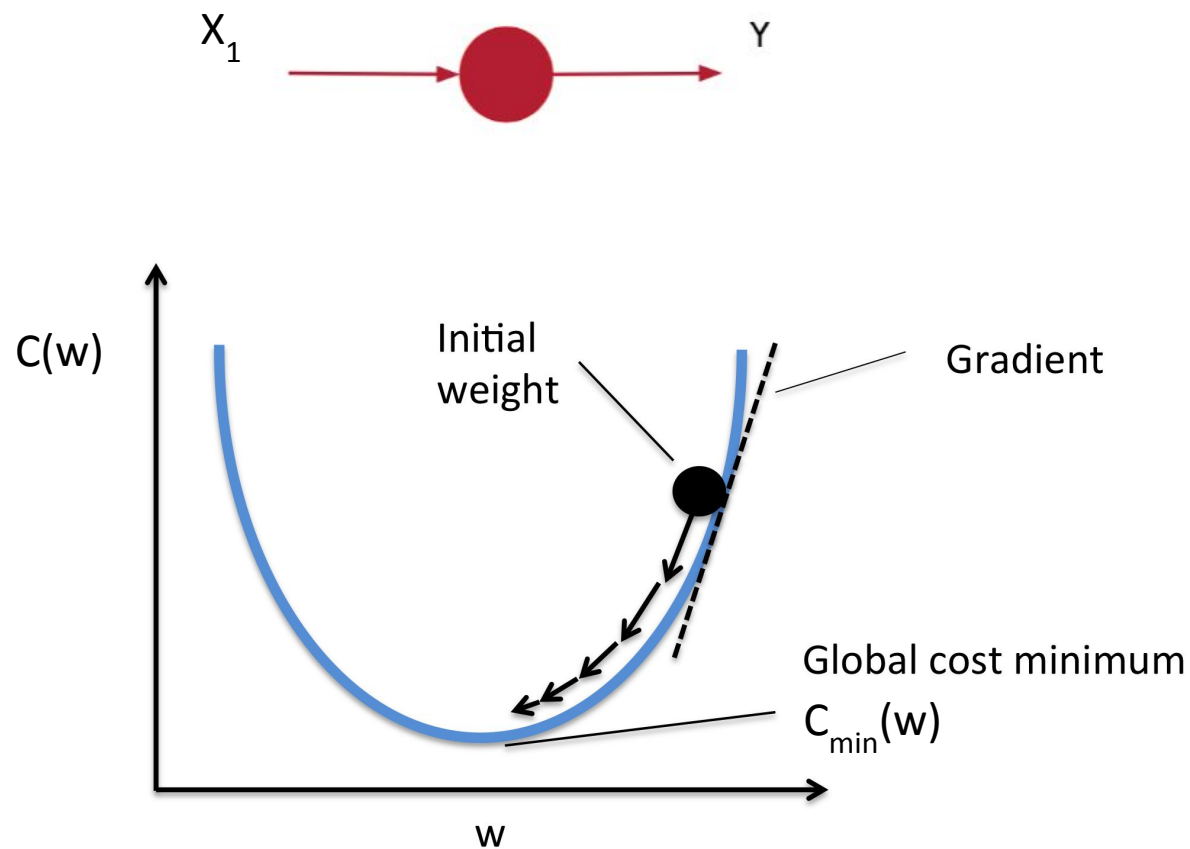
$$Y = w * X + b$$



Cost Function

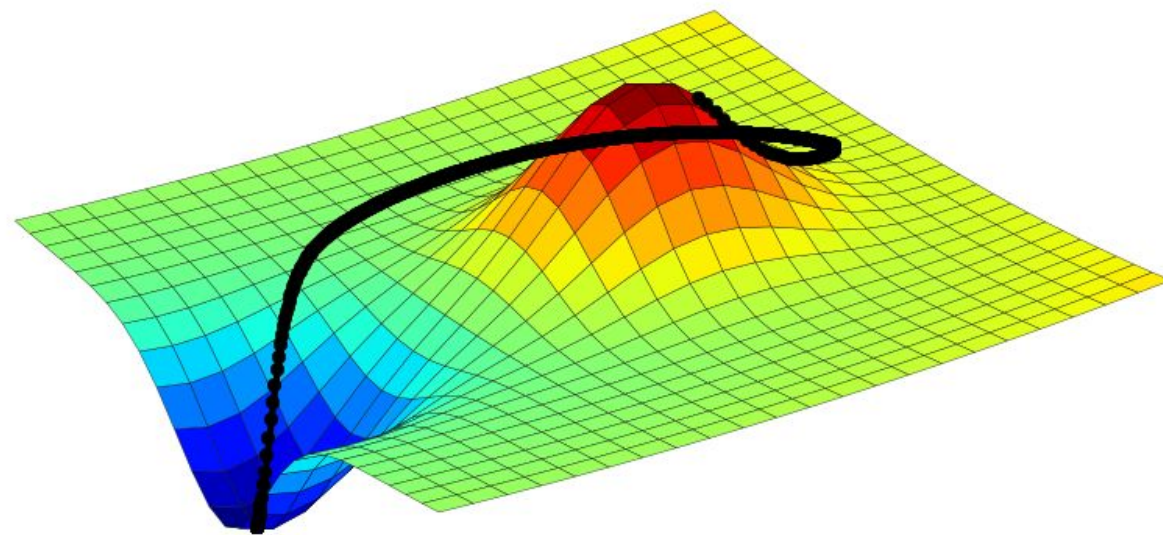
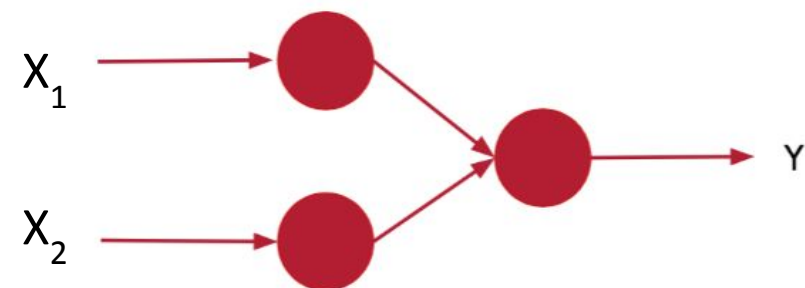
X_1	Y
-1	-3
0	-1
1	1
2	3
3	5
4	7

$$Y = w_1 * X_1 + b_0$$



Cost Function

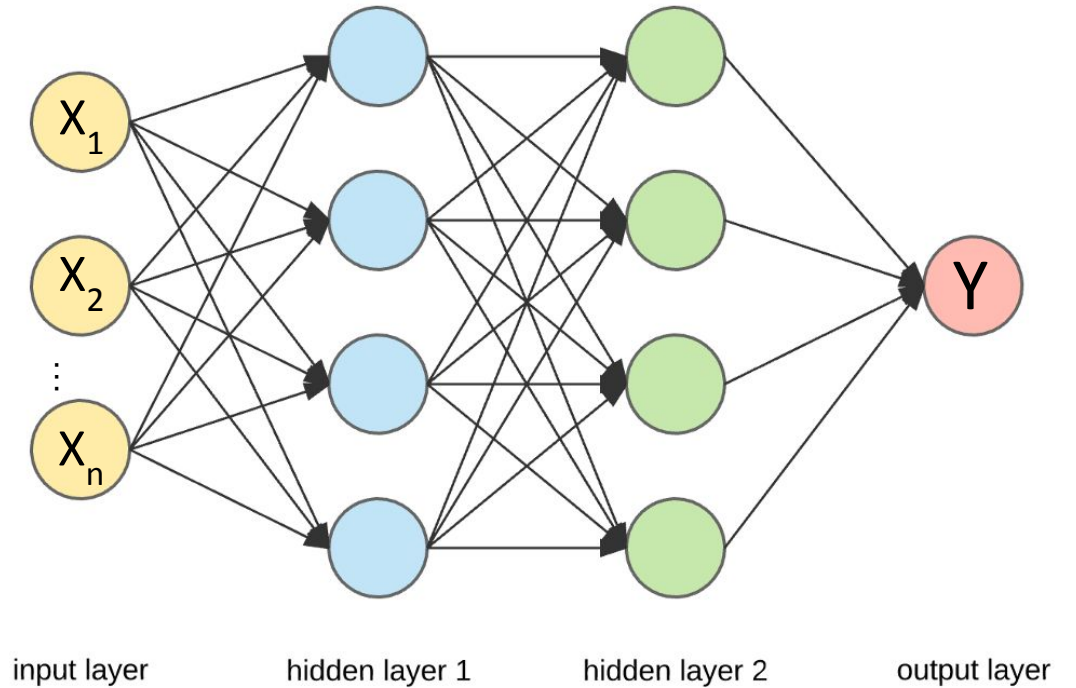
X_1	X_2	Y
-1	-8	-8
0	1	0
1	3	7
2	7	1
3	0	2
4	2	3



$$Y = w_1^* X_1 + w_2^* X_2 + b_u$$

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



$$Y = w_1^* X_1 + w_2^* X_2 + \dots + w_n^* X_n + b_0$$

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()
```

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)
```

Machine Learning Workflow



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

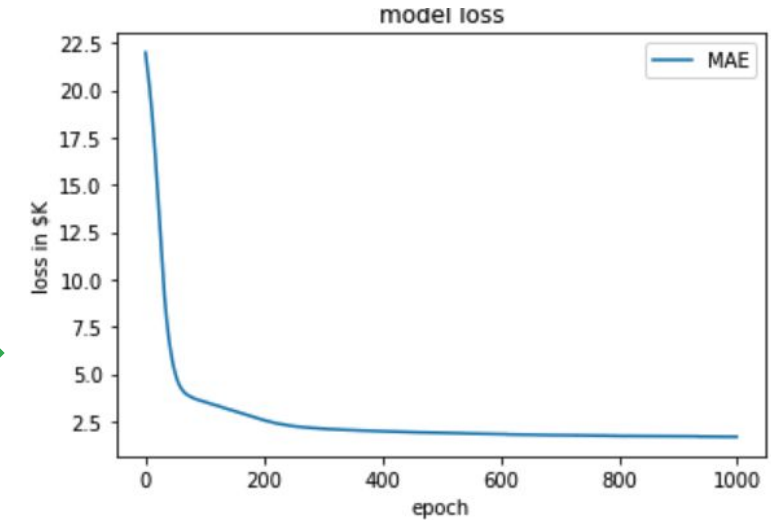
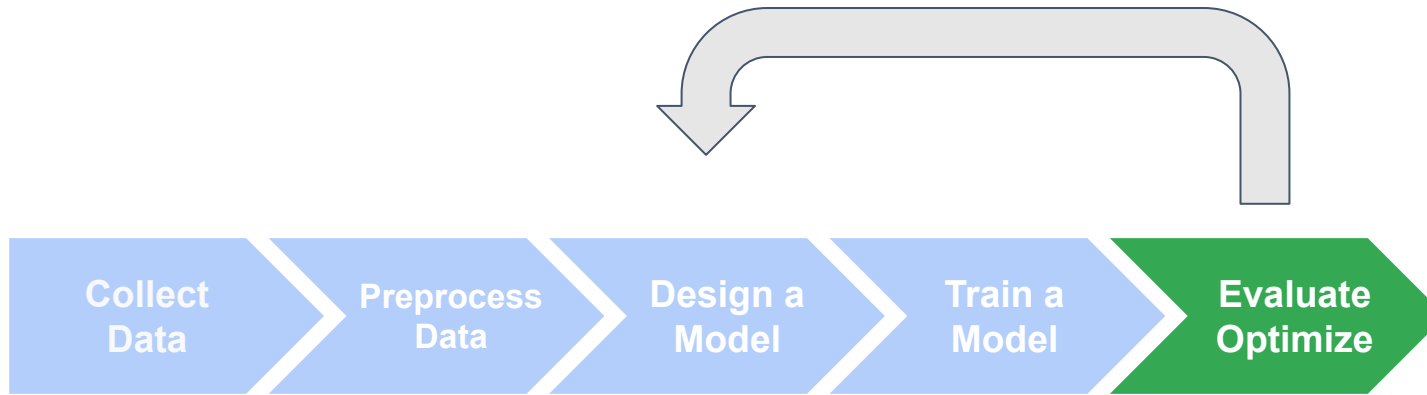
```
model.compile(
    optimizer='adam',
    loss='mse',
    metrics=['mae']
)
```

Machine Learning Workflow



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


Machine Learning Workflow



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

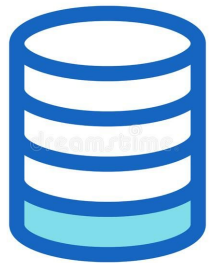
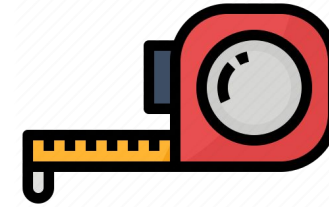
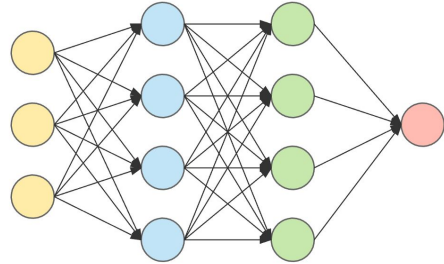
```
tuner.search(
    x_train_norm, y_train,
    epochs=500,
    validation_data=(x_test_norm, y_test))
```

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)
```

Machine Learning Workflow



Reading Material

Main references

- [Harvard School of Engineering and Applied Sciences - CS249r: Tiny Machine Learning](#)
- [Professional Certificate in Tiny Machine Learning \(TinyML\) – edX/Harvard](#)
- [Introduction to Embedded Machine Learning - Coursera/Edge Impulse](#)
- [Computer Vision with Embedded Machine Learning - Coursera/Edge Impulse](#)
- Fundamentals textbook: [“Deep Learning with Python” by François Chollet](#)
- Applications & Deploy textbook: [“TinyML” by Pete Warden, Daniel Situnayake](#)
- Deploy textbook [“TinyML Cookbook” by Gian Marco Iodice](#)

I want to thank **Shawn Hymel** and Edge Impulse, **Pete Warden** and **Laurence Moroney** from Google, Professor **Vijay Janapa Reddi** and **Brian Plancher** from Harvard, and the rest of the **TinyMLedu** team for preparing the excellent material on TinyML that is the basis of this course at UNIFEI.

The IESTI01 course is part of the **TinyML4D**, an initiative to make TinyML education available to everyone globally.

Thanks



UNIFEI