Day 2

Welcome to Day 2 AI Toolkit

Please perform PRE-WORK

1. Access the virtual Lab using link https://html.inspiredvlabs.com Use the username TEKBD142-XX (replace XX with your number) and password

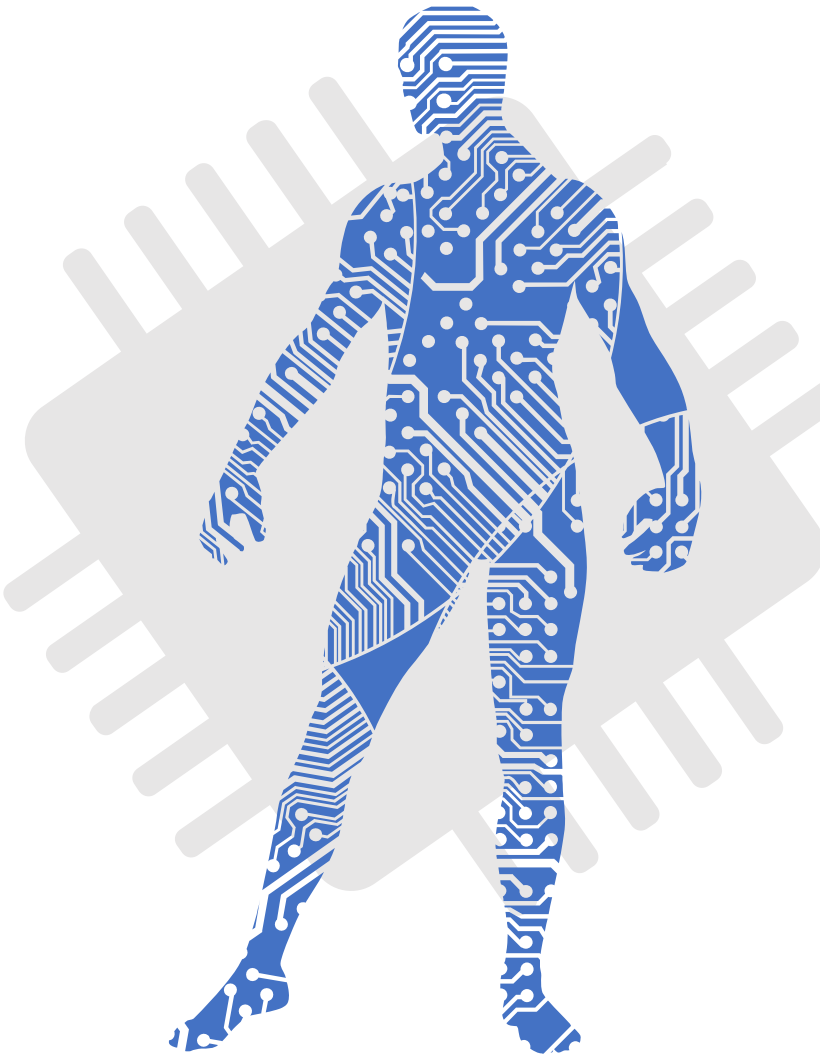
TekBD142!23

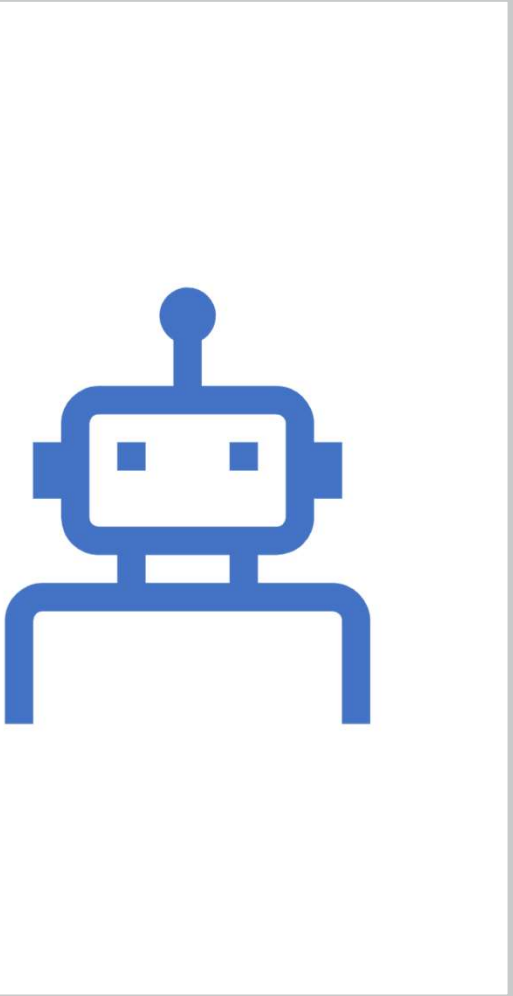
We will be starting soon

Last Name	First Name	Login Id
AHMED	NAZEER	TEKBD142-01
AM	GANESH	TEKBD142-02
BISWAS	SOURAV	TEKBD142-03
CHACKO	THOMAS	TEKBD142-04
JAJOO	SANDESH	TEKBD142-05
MATTOX	CHRISTEL	TEKBD142-06
MAXSON	CRAIG	TEKBD142-07
MURPHY	MATTHEW	TEKBD142-08
PANDIAN	JEYARAJ	TEKBD142-09
PATURI	RAVIKIRAN	TEKBD142-10
RANI	AMITA	TEKBD142-11
SINGLA	SANJEEV	TEKBD142-12
VEERAMASU	BRAHMA RAO	TEKBD142-13

Agenda – Day 2

1. Recap of Day 1
2. Classification
3. Pipeline and Model Persistence
4. Word Cloud
5. TensorFlow
6. Artificial Neural Networks (ANN)
7. Multiple Hands-on





Recap Day - 1

- AI – Machine Learning – Deep Learning
- Machine Learning Introduction
- Machine Learning Techniques
- Machine Learning Development
- Linear Regression
- Model Metrics
- Multiple Hands-on

Categorical Data Handling

DataFrame (some of the columns)

Age	Sex	Race	Income
39	Male	White	<=50K
50	Female	Black	>50K
38	Female	Asian	<=50K
53	Male	Other	>50K

Label Encoder / Imputer

Age	Sex	Race	Income
39	0	0	0
50	1	1	1
38	1	2	0
53	0	3	1

One-Hot Encoding

Age	Sex	Race	Race_ White	Race_ Black	Race_ Asian	Race_ Other	Income
39	0	0	1	0	0	0	0
50	1	1	0	1	0	0	1
38	1	2	0	0	1	0	0
53	0	3	0	0	0	1	1

Column Transformer

Age	Sex	Race	Race_ White	Race_ Black	Race_ Asian	Race_ Other	Income
39	0	0	1	0	0	0	0
50	1	1	0	1	0	0	1
38	1	2	0	0	1	0	0
53	0	3	0	0	0	1	1

Column Transformer:

Age	Sex	Race	Race_ White	Race_ Black	Race_ Asian	Race_ Other	Income
39	0	0	1	0	0	0	0
50	1	1	0	1	0	0	1
38	1	2	0	0	1	0	0
53	0	3	0	0	0	1	1

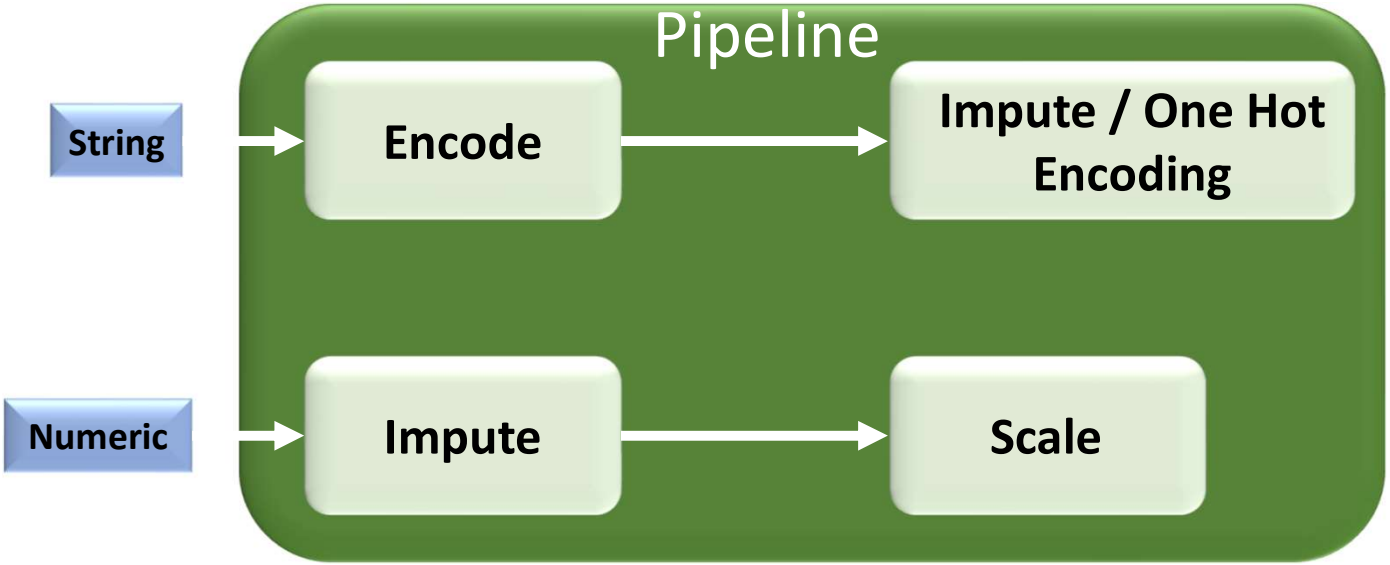
Column Transformer

Age	Sex	Race	Race_ White	Race_ Black	Race_ Asian	Race_ Other	Income
39	0	0	1	0	0	0	0
50	1	1	0	1	0	0	1
38	1	2	0	0	1	0	0
53	0	3	0	0	0	1	1

Column Transformer:

Age	Sex	Race	Race_ White	Race_ Black	Race_ Asian	Race_ Other	Income
39	0	0	1	0	0	0	0
50	1	1	0	1	0	0	1
38	1	2	0	0	1	0	0
53	0	3	0	0	0	1	1

Pipeline



- A machine learning work-flow
- Made up of number of stages
- Can be persisted

Logistic Regression

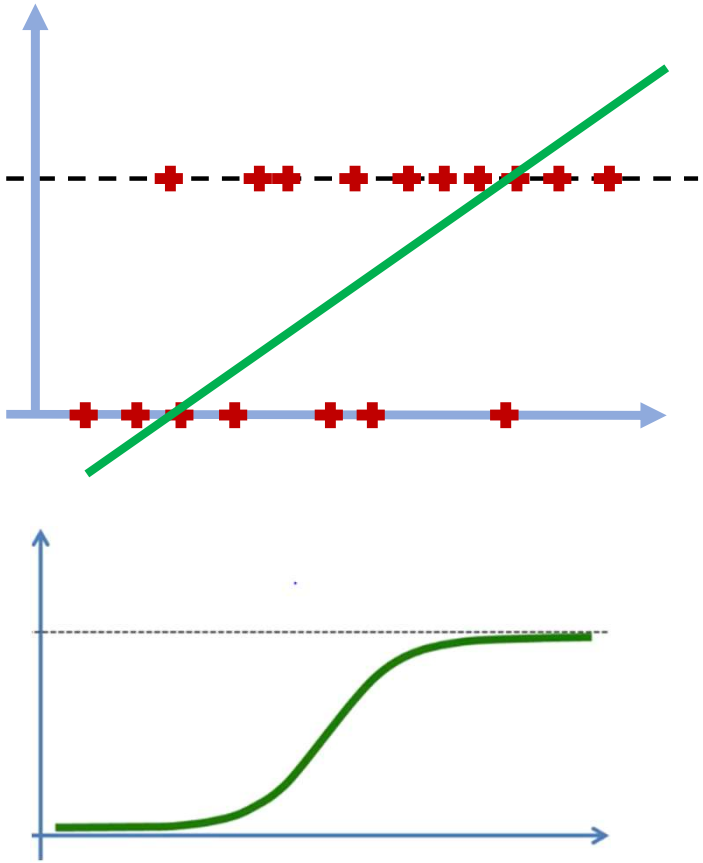
Logistic Regression

$$y = b_0 + b_1 * x_1$$

Sigmoid Function

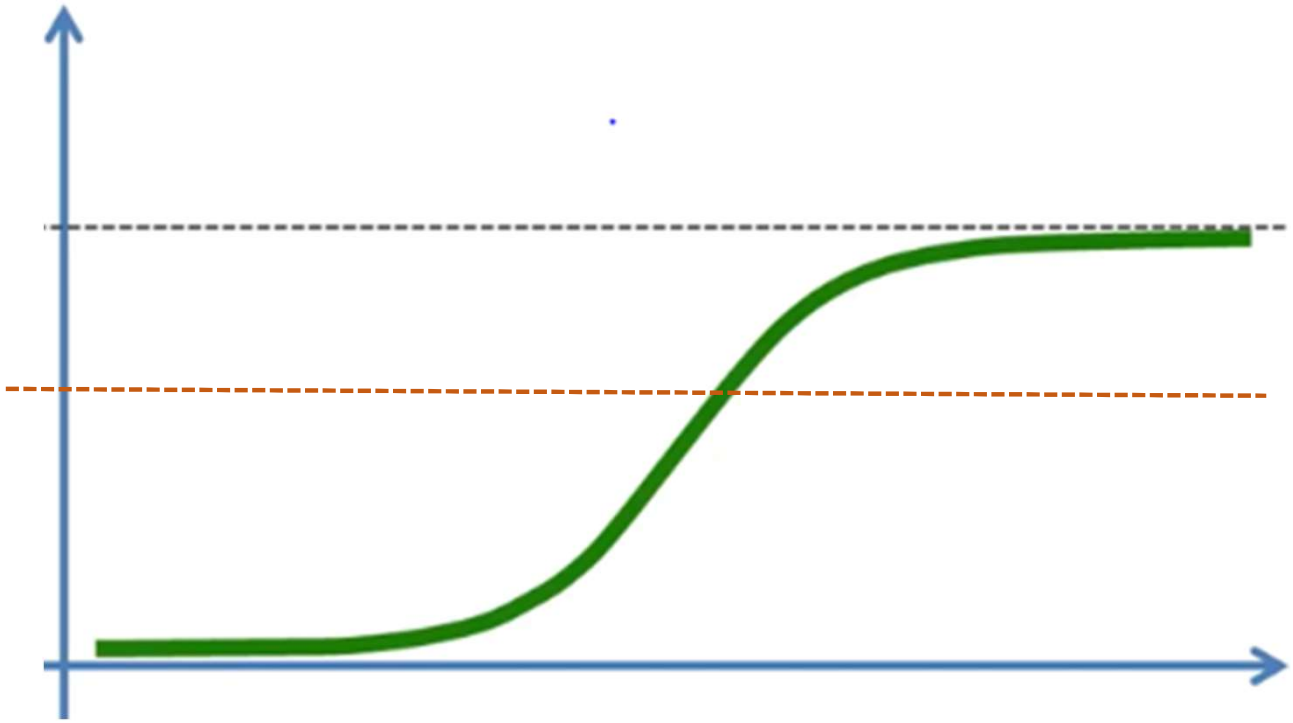
$$p = \frac{1}{1 + e^{-y}}$$

$$\ln\left(\frac{p}{1-p}\right) = b_0 + b_1 * x$$



This is the formula for logistic regression

Logistic Regression



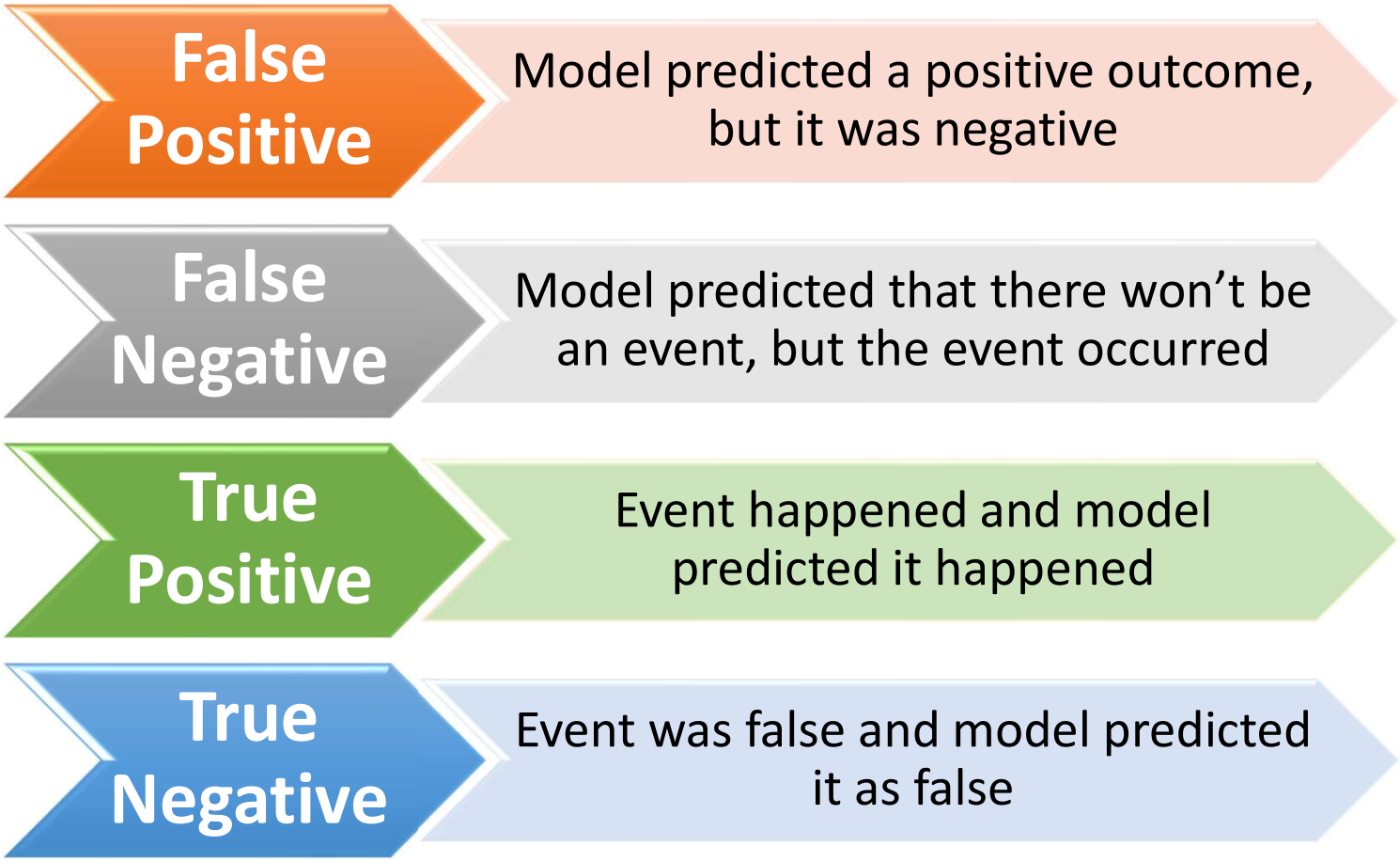
positive class = 1

negative class = 0

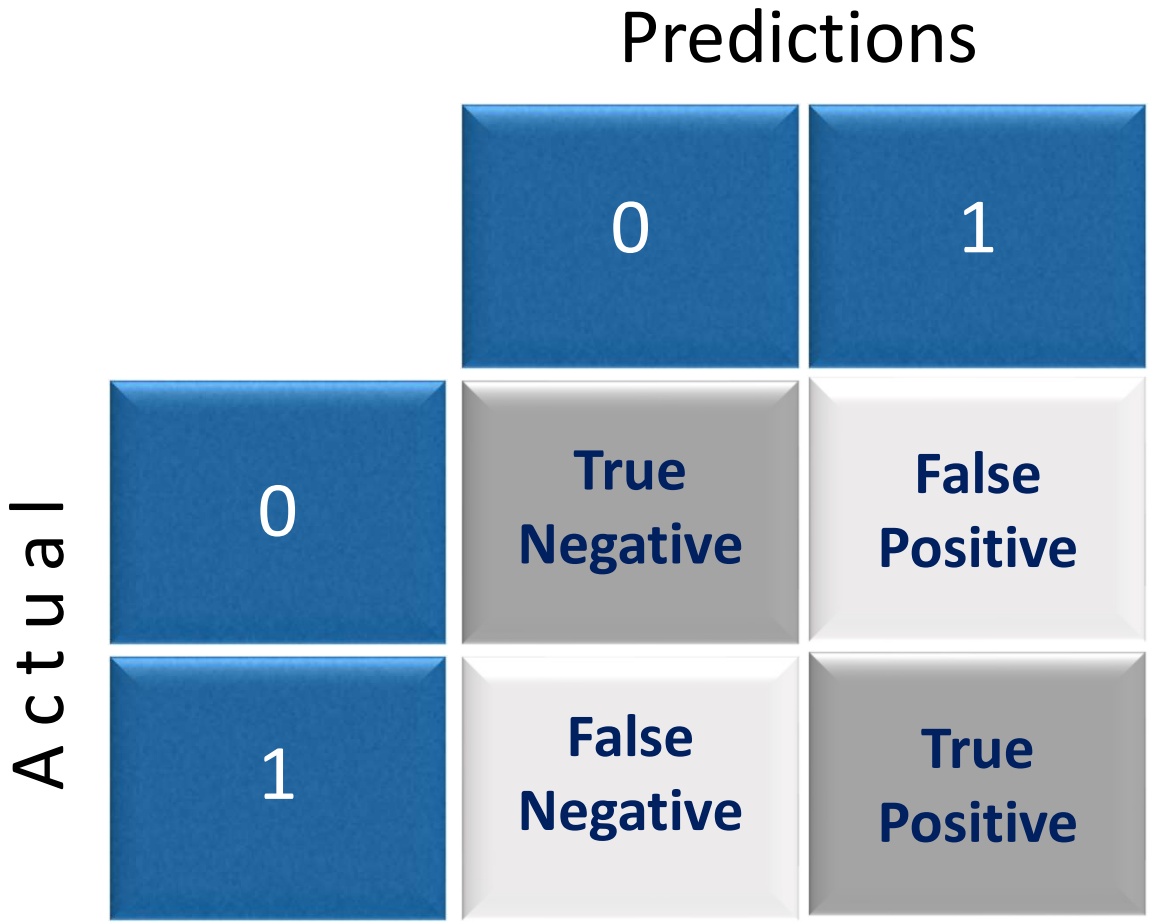
For every class, the model gives probability

Model Metrics

Model Evaluation



Confusion Matrix



$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Classification Report

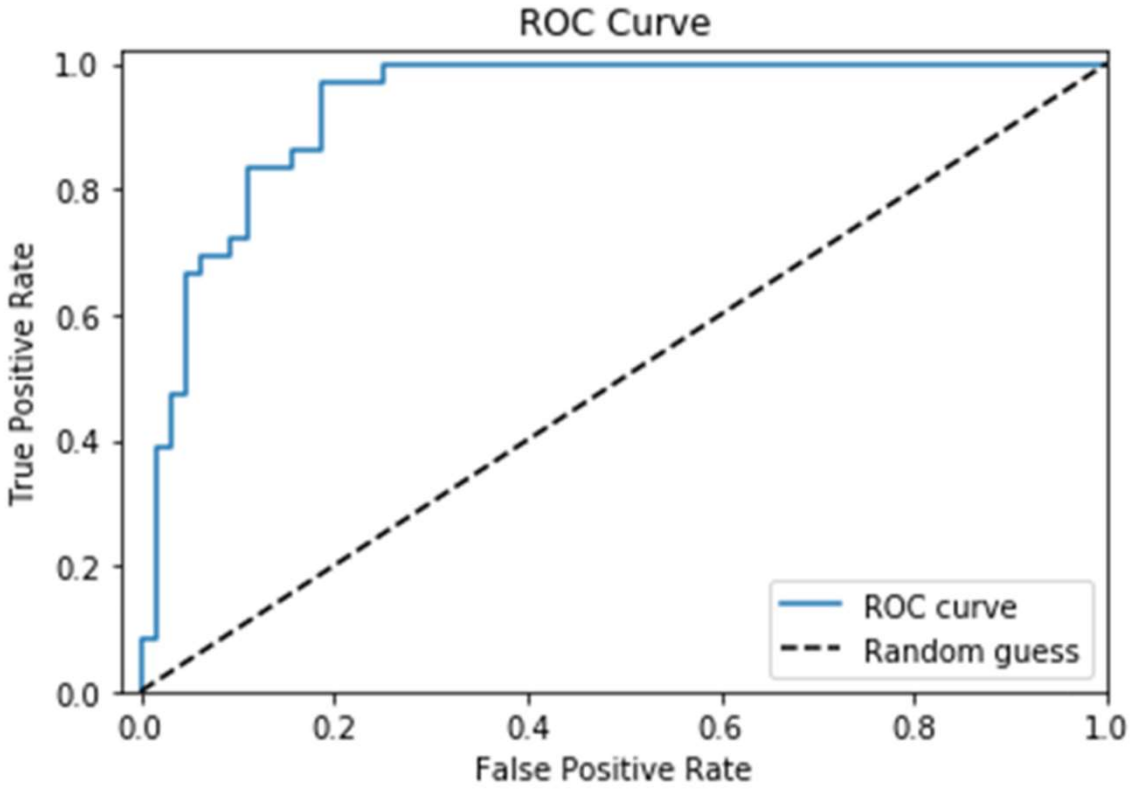
	precision	recall	f1-score	support
0	0.85	0.94	0.89	64
1	0.86	0.69	0.77	36

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$f1score = 2 \times \frac{precision \times recall}{precision + recall}$$

ROC Curve (Receiver Operating Characteristic)



Model Persistence

Model Persistence

Once the model is built, want to persist it so that the model can be used to make predictions for "new" data

Such persisted model is loaded by another script that uses it to make predictions

Not only the model needs to be persisted but also any scalers or pipelines used on data pre-processing

Persistence

Persistence can be applied to Models, Pipelines, Scalers

Pickle

```
from pickle import dump
# save model (in script 1)
dump(model, open('filename.pkl', 'wb')

# load model (in script 2)
from pickle import load
aModel = load(open('filename.pkl', 'rb'))
aModel.predict(data)
```

Joblib

```
from joblib import dump
# save model (in script 1)
dump(model, open('filename.joblib'))

# load model (in script 2)
from joblib import load
aModel = load(open('filename.joblib'))
aModel.predict(data)
```

https://scikit-learn.org/stable/modules/model_persistence.html

Census Income Dataset (UCI)

- Continuous – numerical values, rest are all categorical values

- **>50K, <=50K.**
- **age:** continuous.
- **workclass:** Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
- **fnlwgt:** continuous.
- **education:** Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
- **education-num:** continuous.
- **marital-status:** Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
- **occupation:** Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
- **relationship**: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- **race:** White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- **sex:** Female, Male.
- **capital-gain:** continuous.
- **capital-loss:** continuous.
- **hours-per-week:** continuous.
- **native-country:** United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

Logistic Regression

- Open file '**CodeSamples/Logistic Regression**' using Jupyter
- Load agent.csv into DataFrame
- Build a pipeline to handle categorical and numeric data
- Build a Logistic Regression model
- Check performance of model on training dataset
- Make predictions
- Compare predications v/s the actual values
- Persist the pipeline and the model

Load and Predict

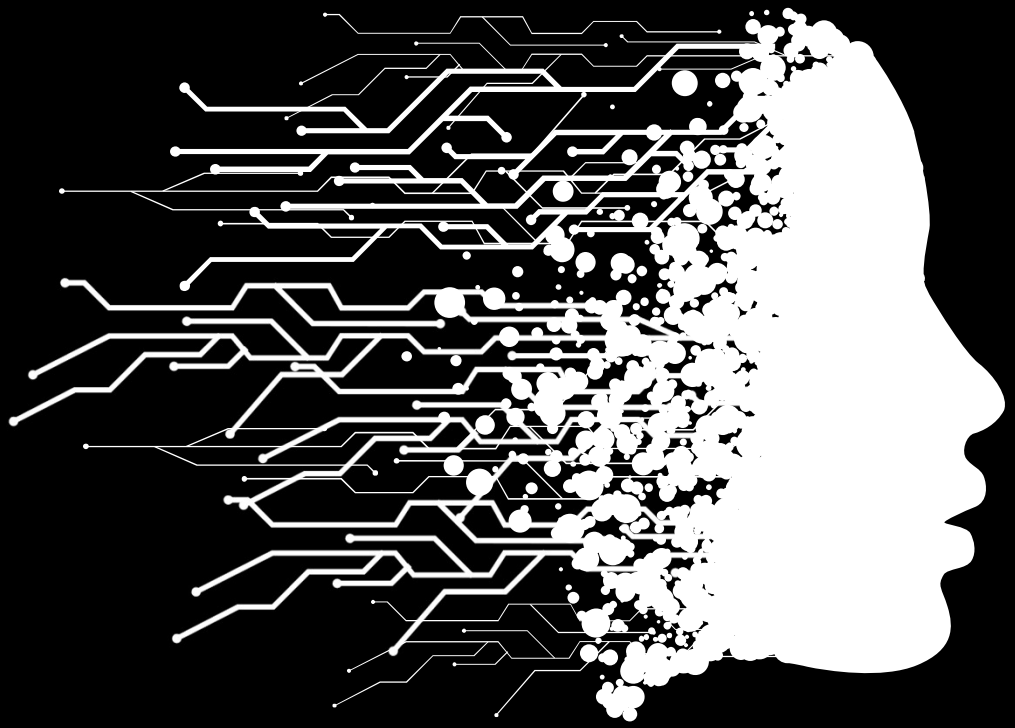
- Open file '**CodeSamples/LoadModel-Predict**' using Jupyter
- Load agent-new.csv into DataFrame (note: no target variable – predicting it)
- Use the pipeline and model persisted in the previous example to apply to new data
- Apply them to new dataframe
- Make predictions and write the predictions to a .csv file

Word Cloud

- Datasets often have many text fields e.g. names, cities, states, descriptions, etc.
- Want to most prominent words
- Word Cloud (python-based library)
  - Get visual representation of the text – most frequent words
  - These are represented in big fonts and in different colors
  - Words in small fonts indicate that the words are not important

Word Cloud

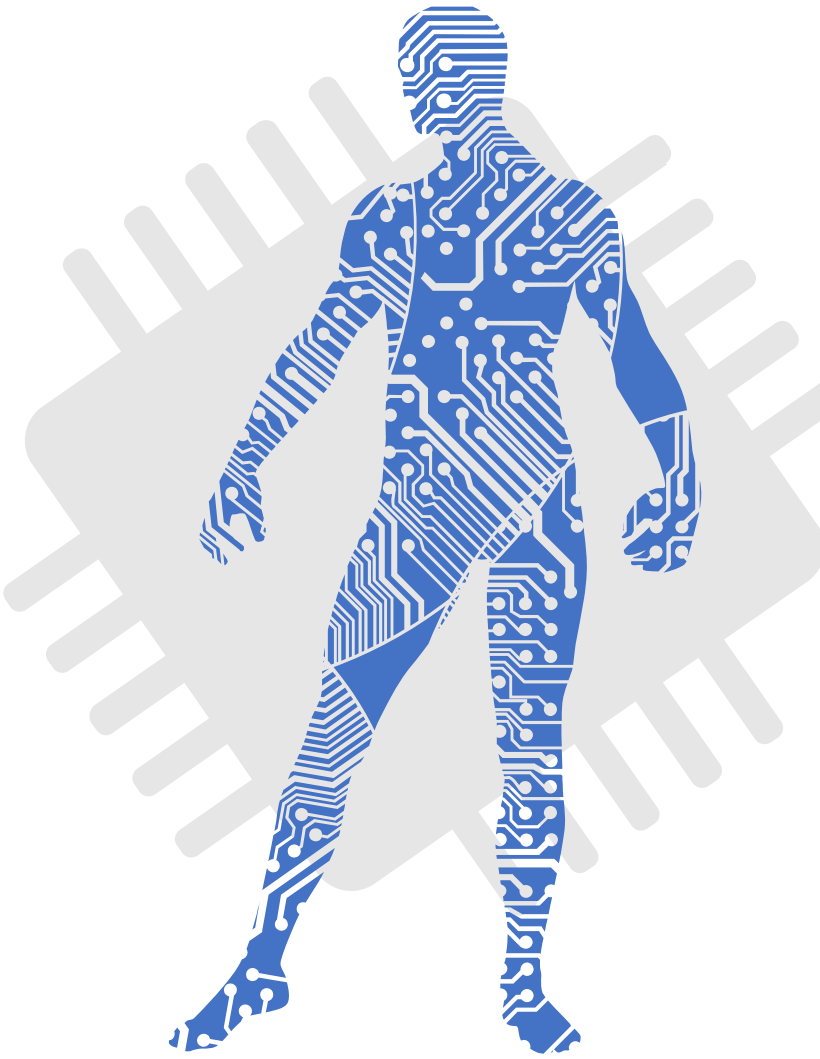
- Open file '**CodeSamples/PetFinder**' using Jupyter
- Dataset consists information about pets (cats, dogs) that are up for adoption
- Use wordcloud to explore the dataset



Deep Learning

Deep Learning

1. Deep Learning Packages
2. Understanding TensorFlow
3. Artificial Neural Networks

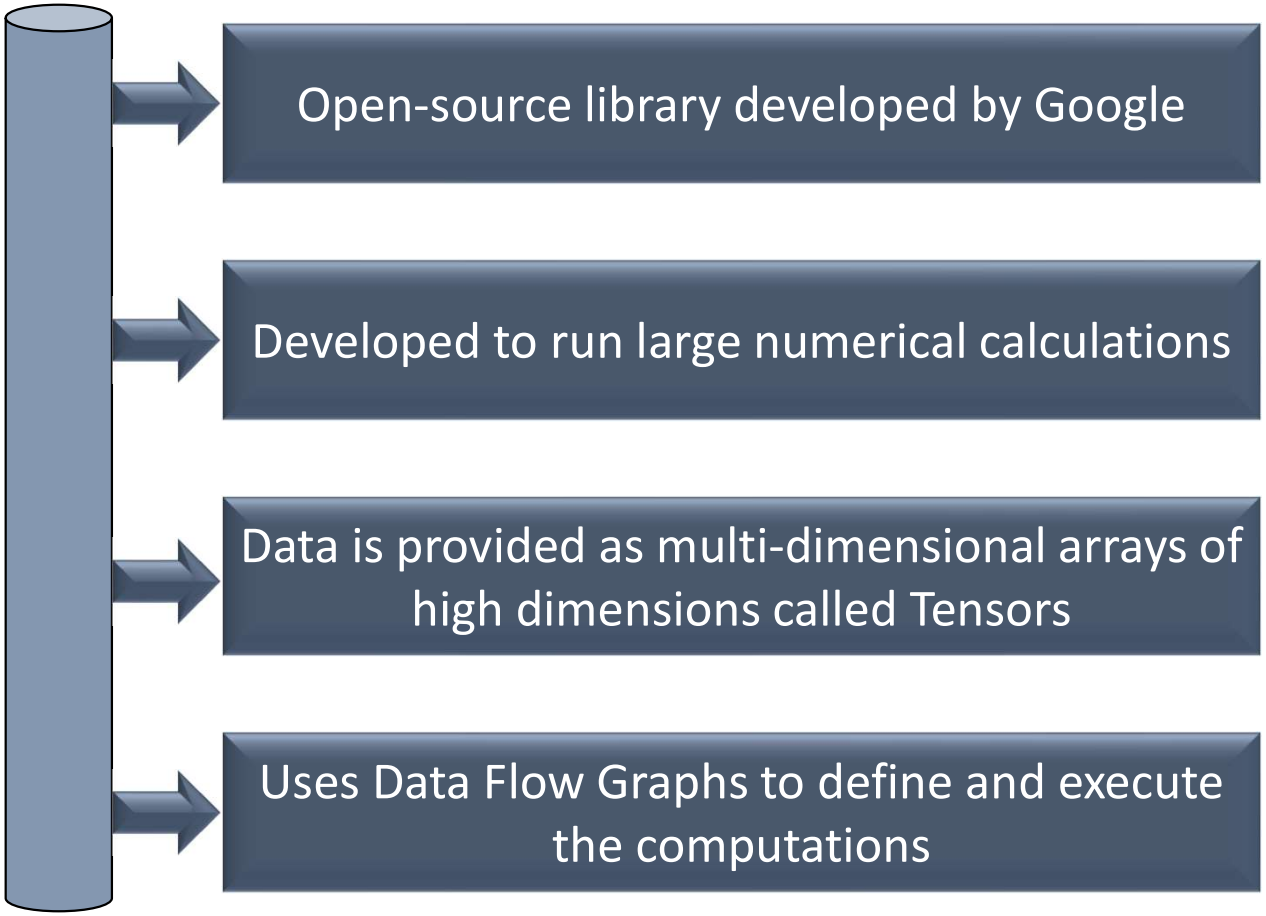


Deep Learning Packages

Deep Learning Package Zoo

Understanding TensorFlow

TensorFlow


Open-source library developed by Google
Developed to run large numerical calculations
Data is provided as multi-dimensional arrays of high dimensions called Tensors
Uses Data Flow Graphs to define and execute the computations

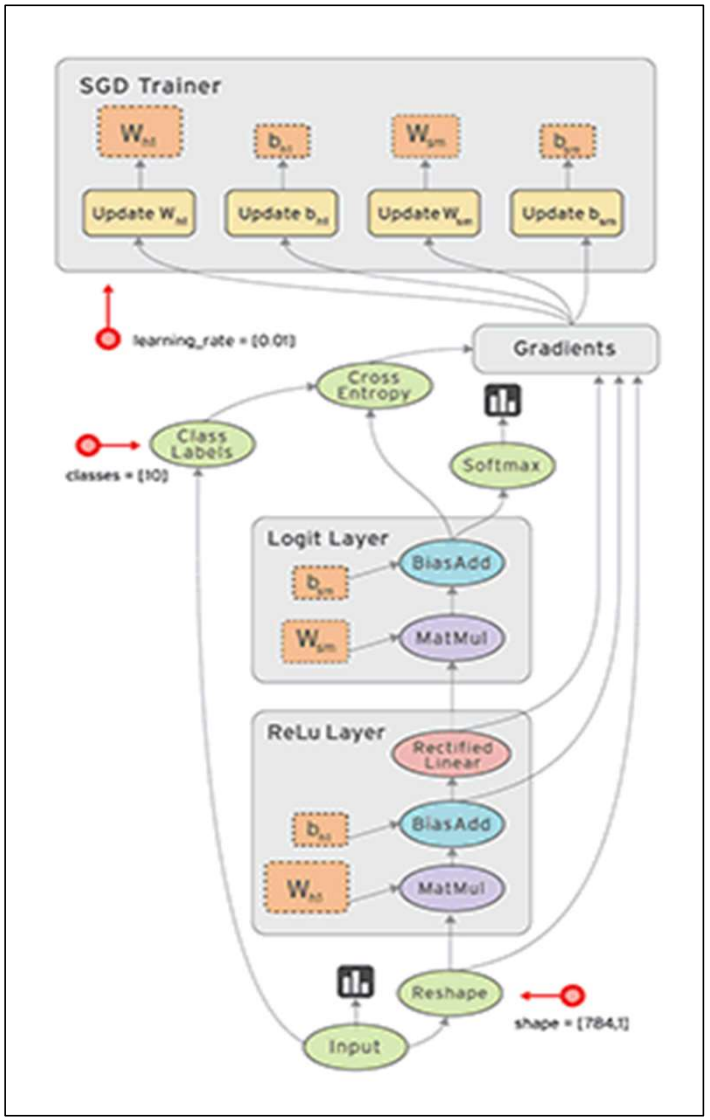
Tensors

Dimension [5,]

Dimension [5, 4]

Dimension [3,3,3]

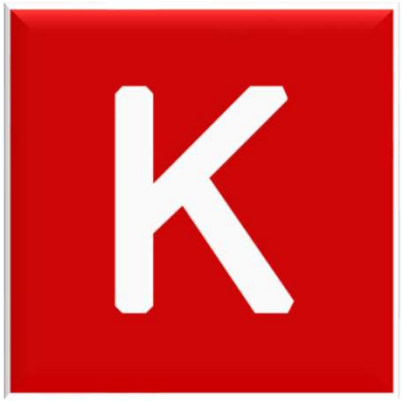
Data Flow Graph ("Flow" in TensorFlow)



Two Basic Steps in TensorFlow

Build Computational Graph

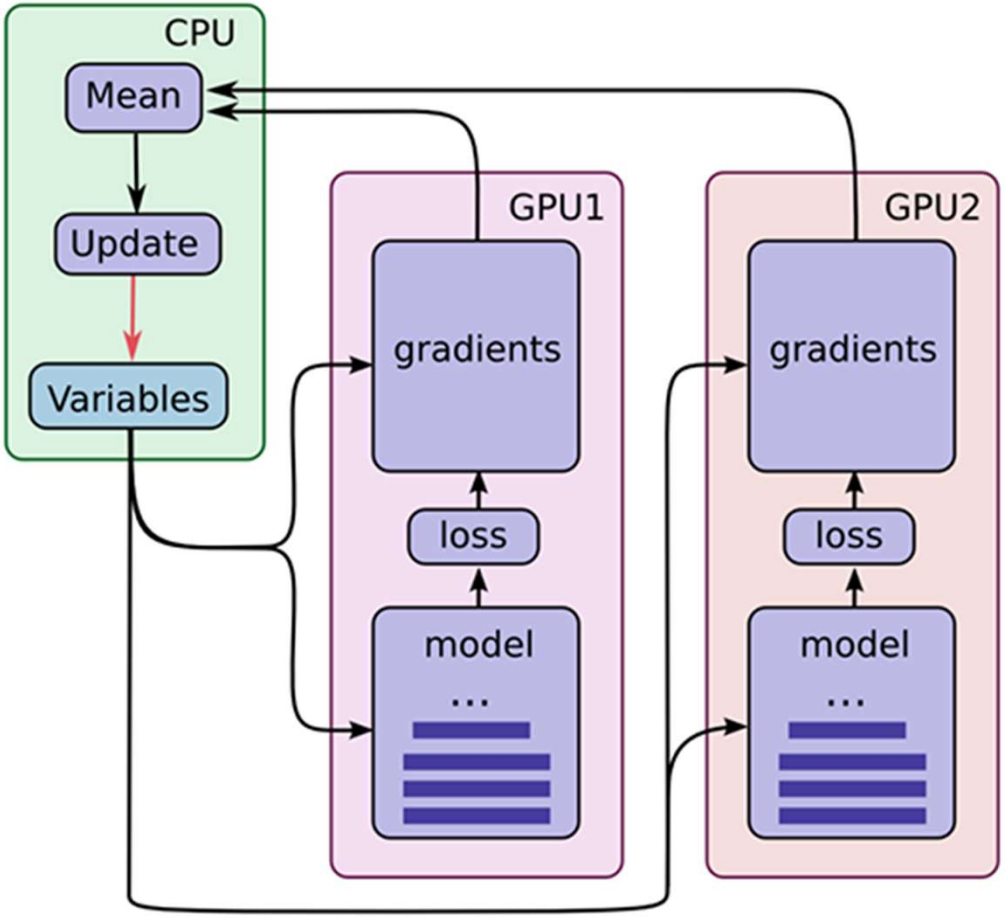
Execute the Computational Graph



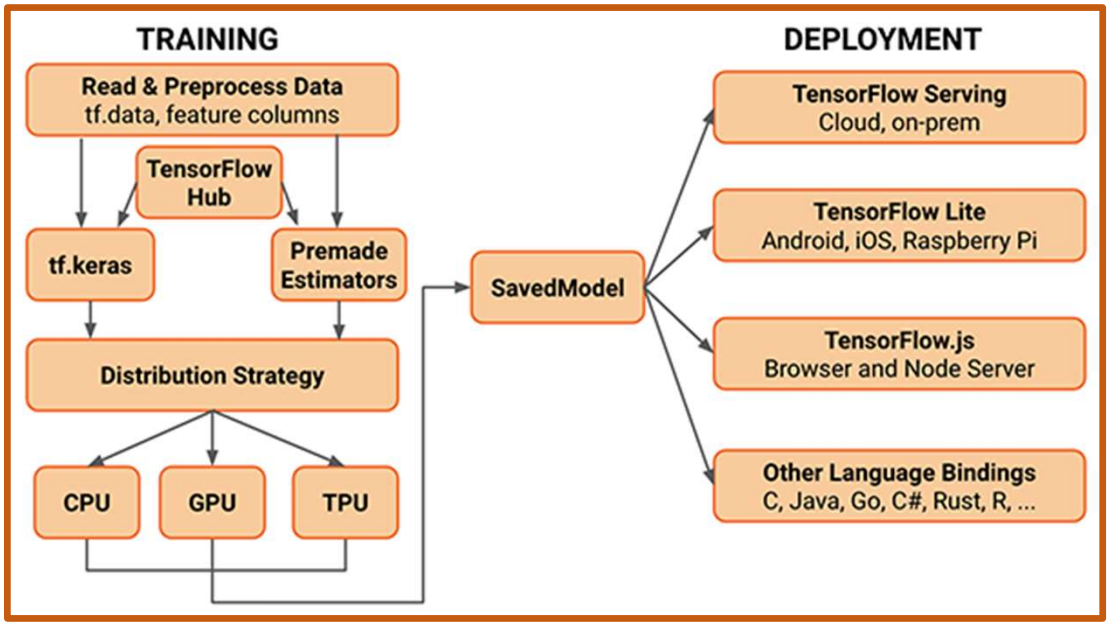
Keras - Backends

TensorFlow: Multi GPU

https://www.tensorflow.org/guide/distributed_training#mirroredstrategy



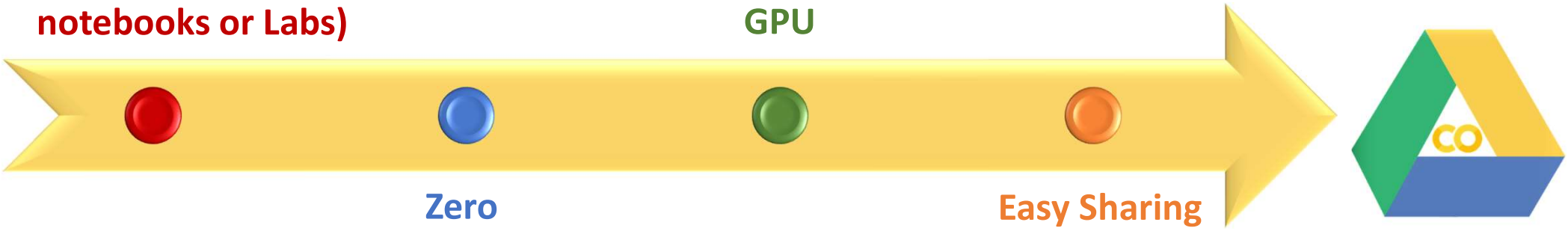
TensorFlow Ecosystem



Google's Colaboratory (Colab)

Colaboratory allows to write Python code in browser in a notebook format (Juypter notebooks or Labs)

Free access to GPU



Zero configuration

Easy Sharing

https://www.youtube.com/watch?v=inN8seMm7UI

Tensors – Colab

- Will be using the Google's Colaboratory
- "Notebook" is used as an IDE
- Looking at basic code for

https://colab.research.google.com/

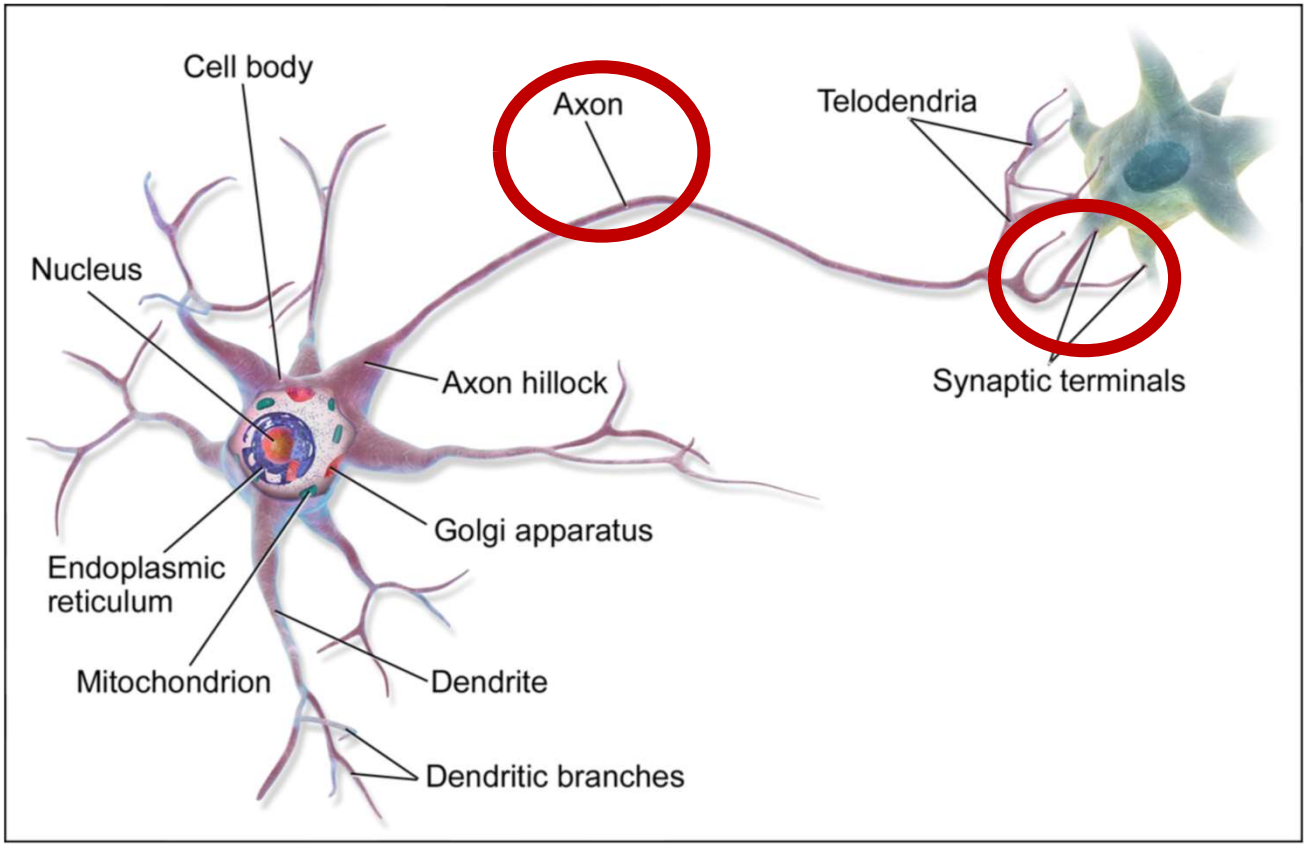
- Make sure to sign in with your gmail
- From the File menu, select "Upload notebook"
- Select the from the Labs folder, "Tensor Introduction.ipynb"

TensorFlow – Tensor Examples

- Open "**CodeSamples/Tensor Introduction**"
- Examples of properties of Tensor such as:
  - Different ranks
  - Different shape
  - Indexing
  - Reshaping

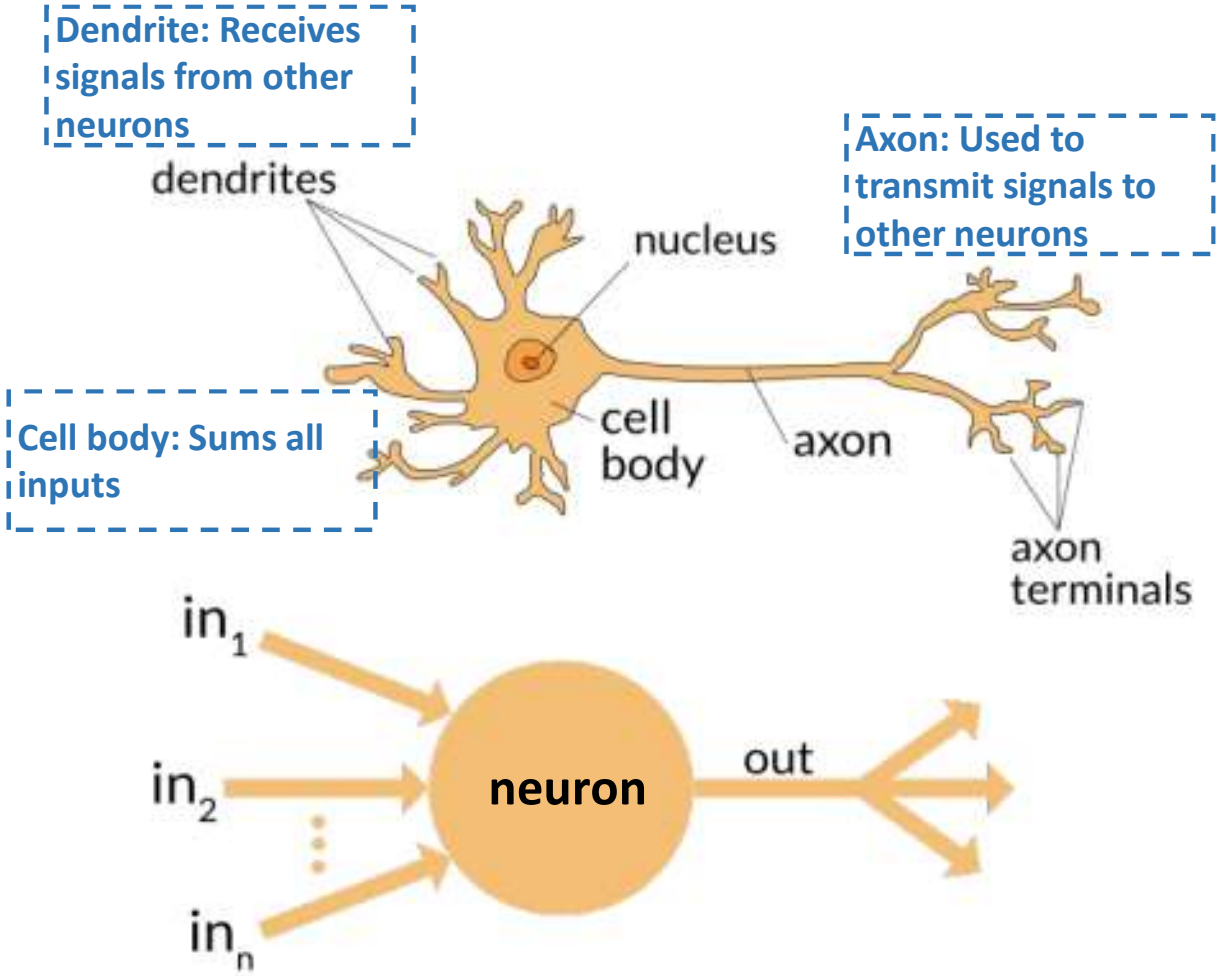
Artificial Neural Networks (ANN)

Neuron

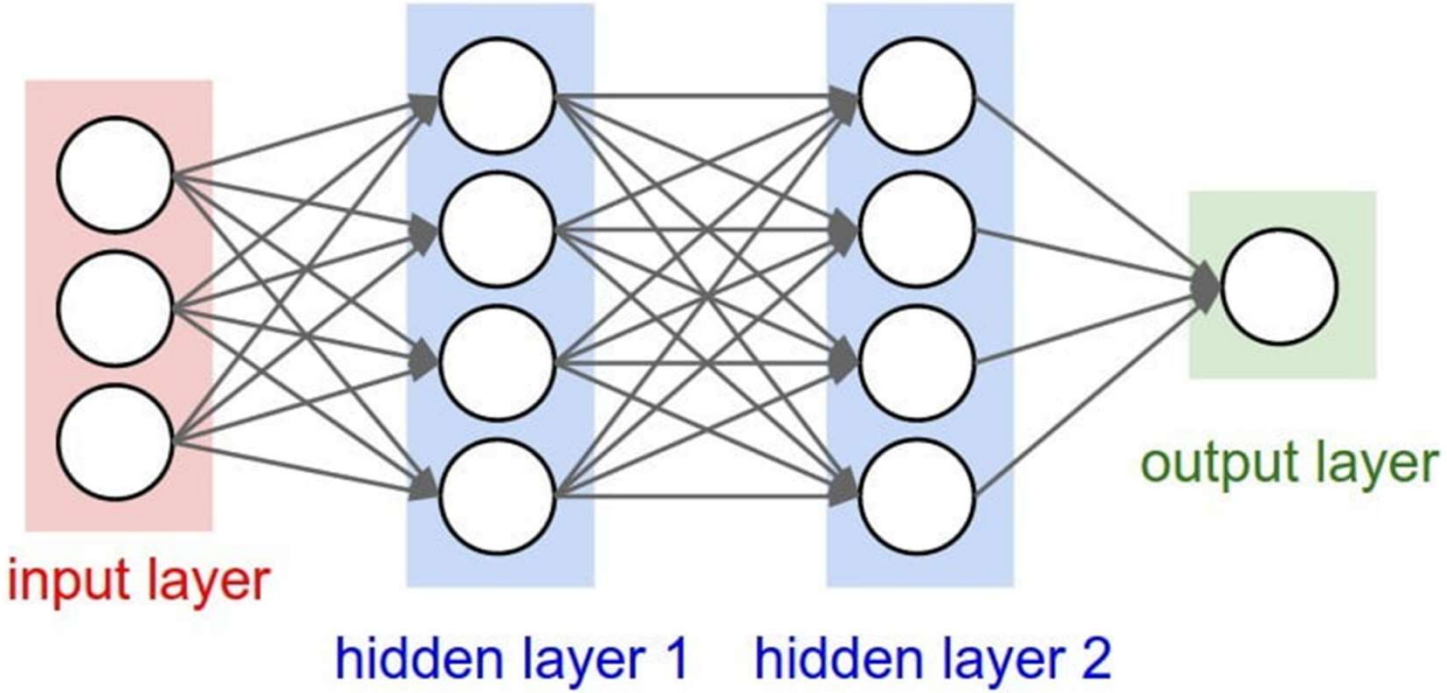


From Wikipedia

Neuron in Artificial Neural Networks



Artificial Neural Network (ANN)



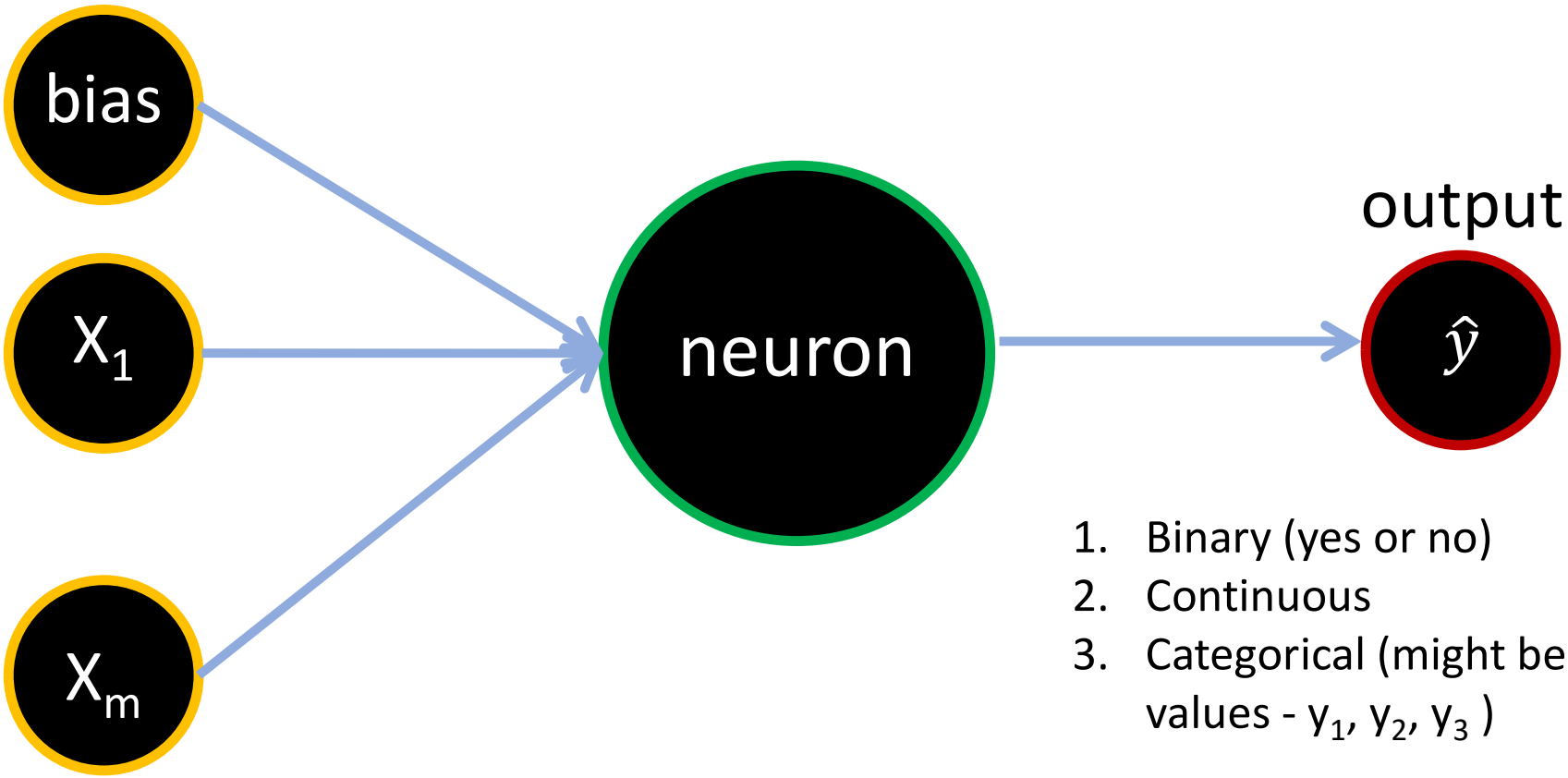
Artificial Neural Networks - Layers

- Each of the layers in ANN perform different tasks
- Some are better suited for certain tasks compared to others

- Types of Layers
  - Dense (or fully connected layers)
  - Convolutional (Image processing)
  - Recurrent (Time series processing)
  - Pooling (Image processing)
  - Many more ….

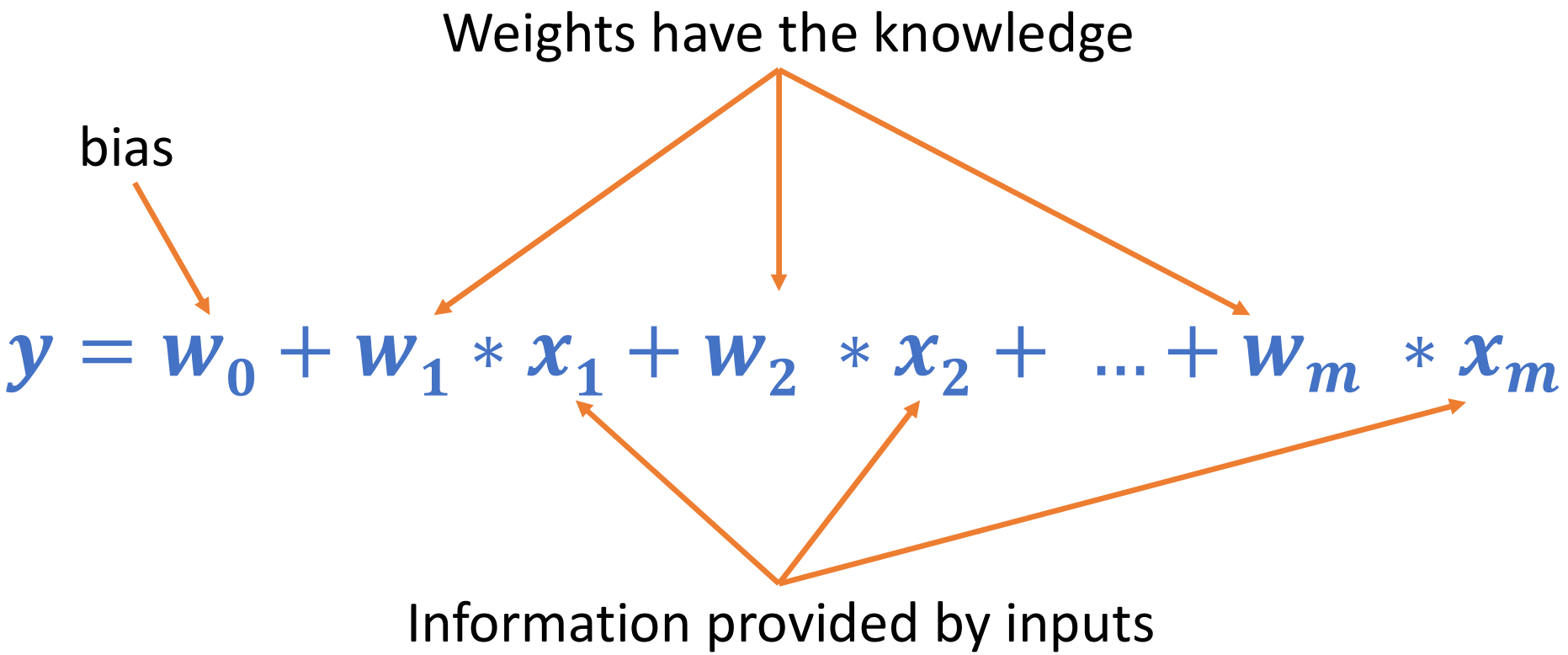
```
# Create Neural Network

model = keras.models.Sequential()
model.add(keras.layers.Flatten(input_shape=[28, 28]))
model.add(keras.layers.Dense(128, activation="relu"))
model.add(keras.layers.Dense(10, activation="softmax"))
```

Artificial Neural Networks - Perceptron

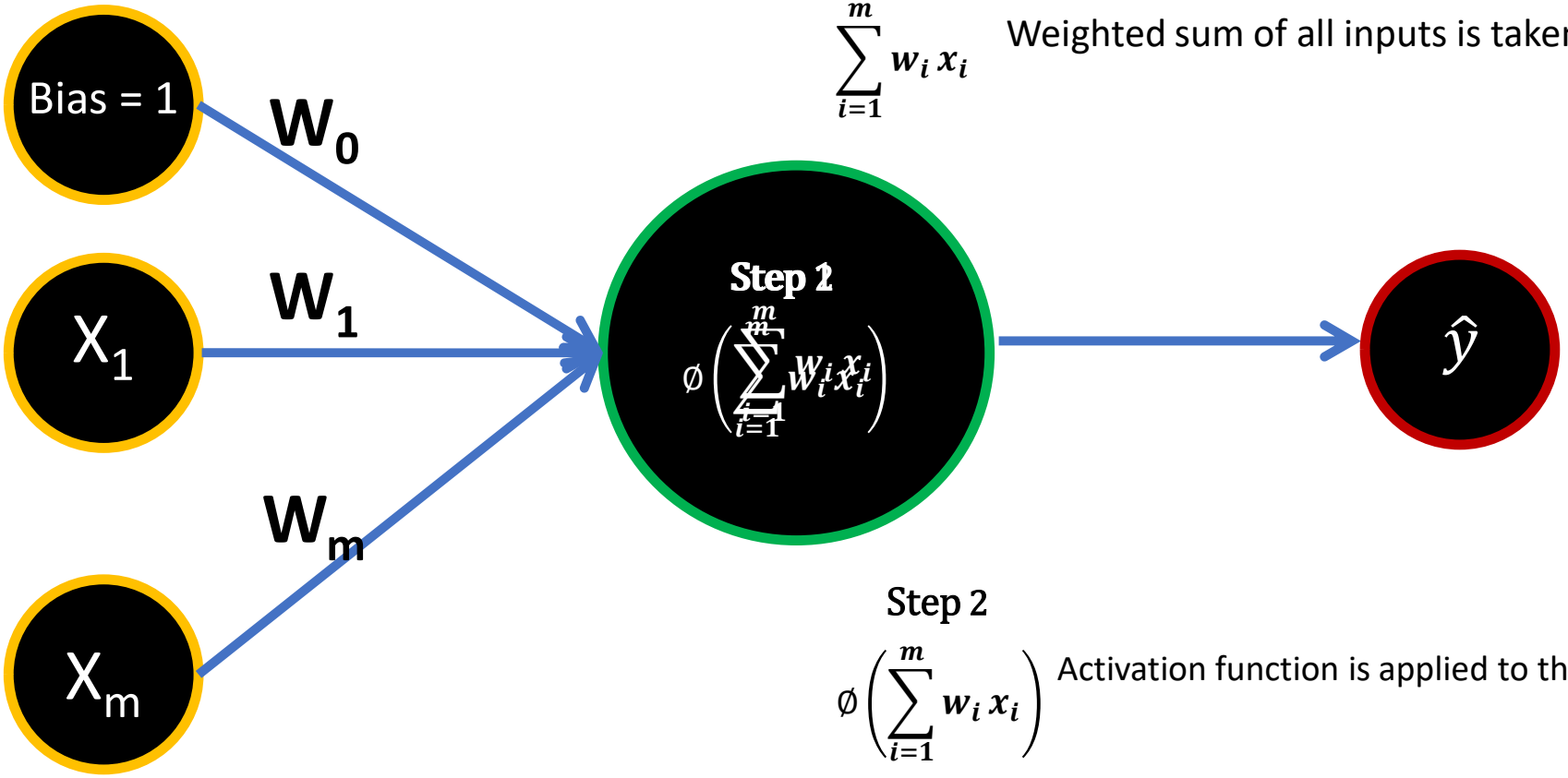


1. Binary (yes or no)
2. Continuous
3. Categorical (might be multiple values - y_1, y_2, y_3)

Weighted Combination of Features



Neural Networks – Weighted Sum



Step 1

$$\sum_{i=1}^{m} w_i x_i$$

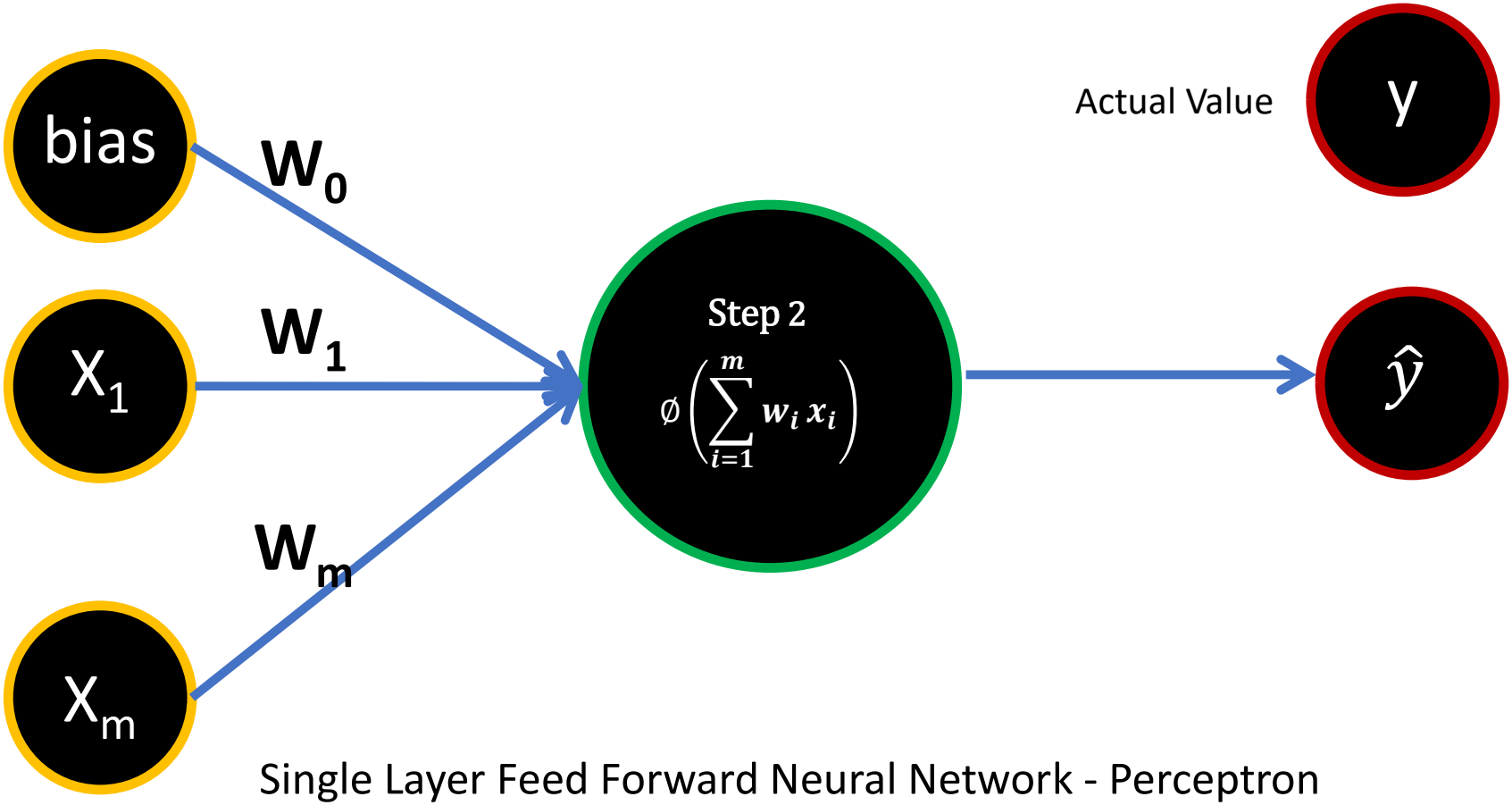
Weighted sum of all inputs is taken

Step 2

$$\emptyset\left(\sum_{i=1}^{m} w_i x_i\right)$$

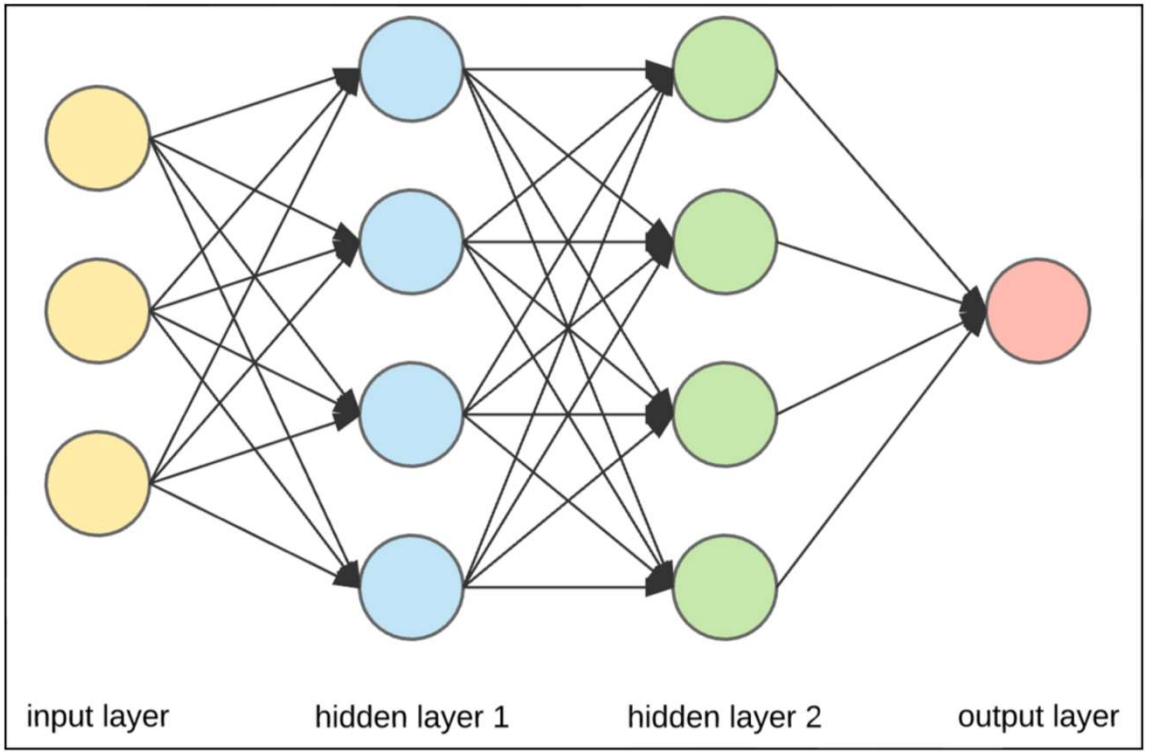
Activation function is applied to the sum

Single Layer Perceptron

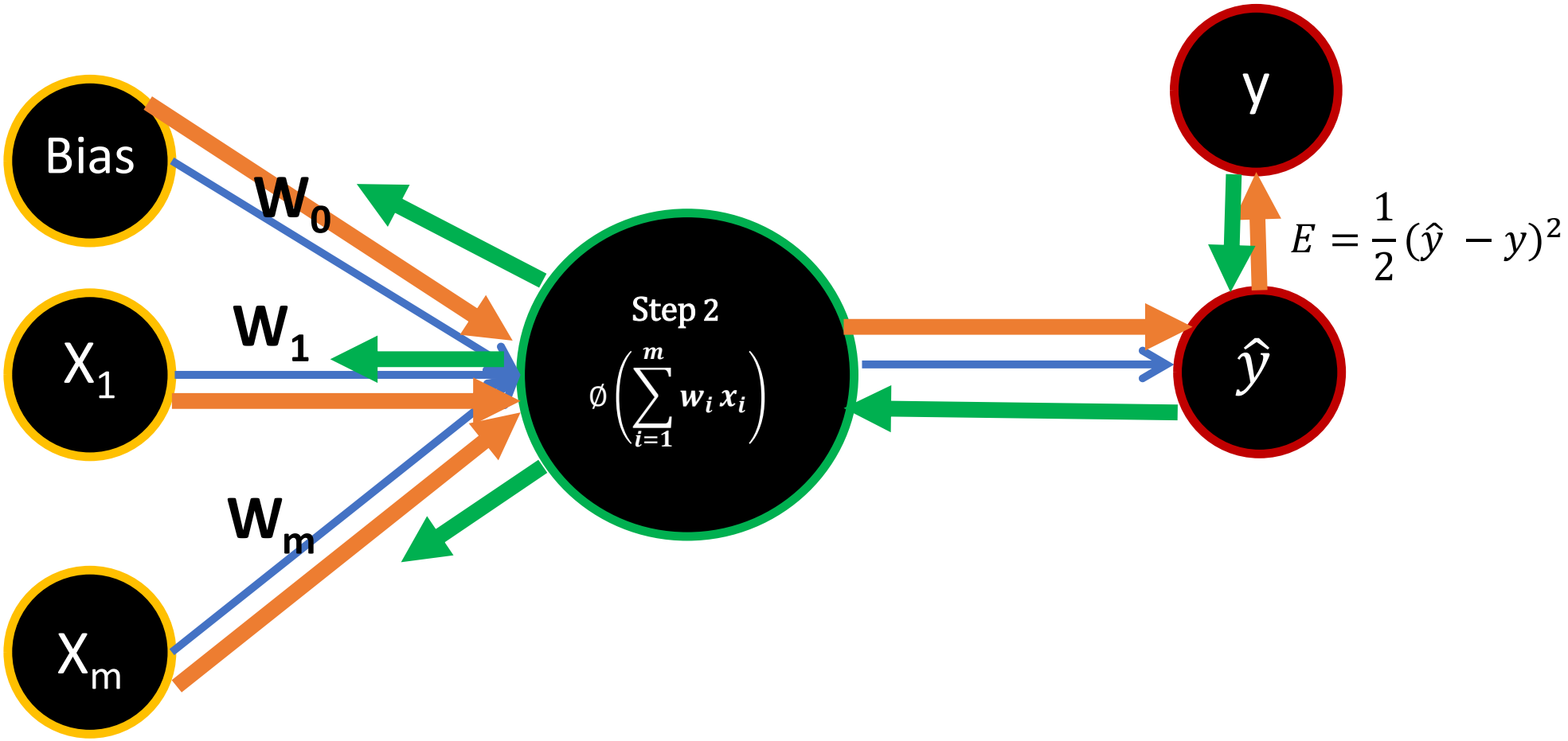


Single Layer Feed Forward Neural Network - Perceptron

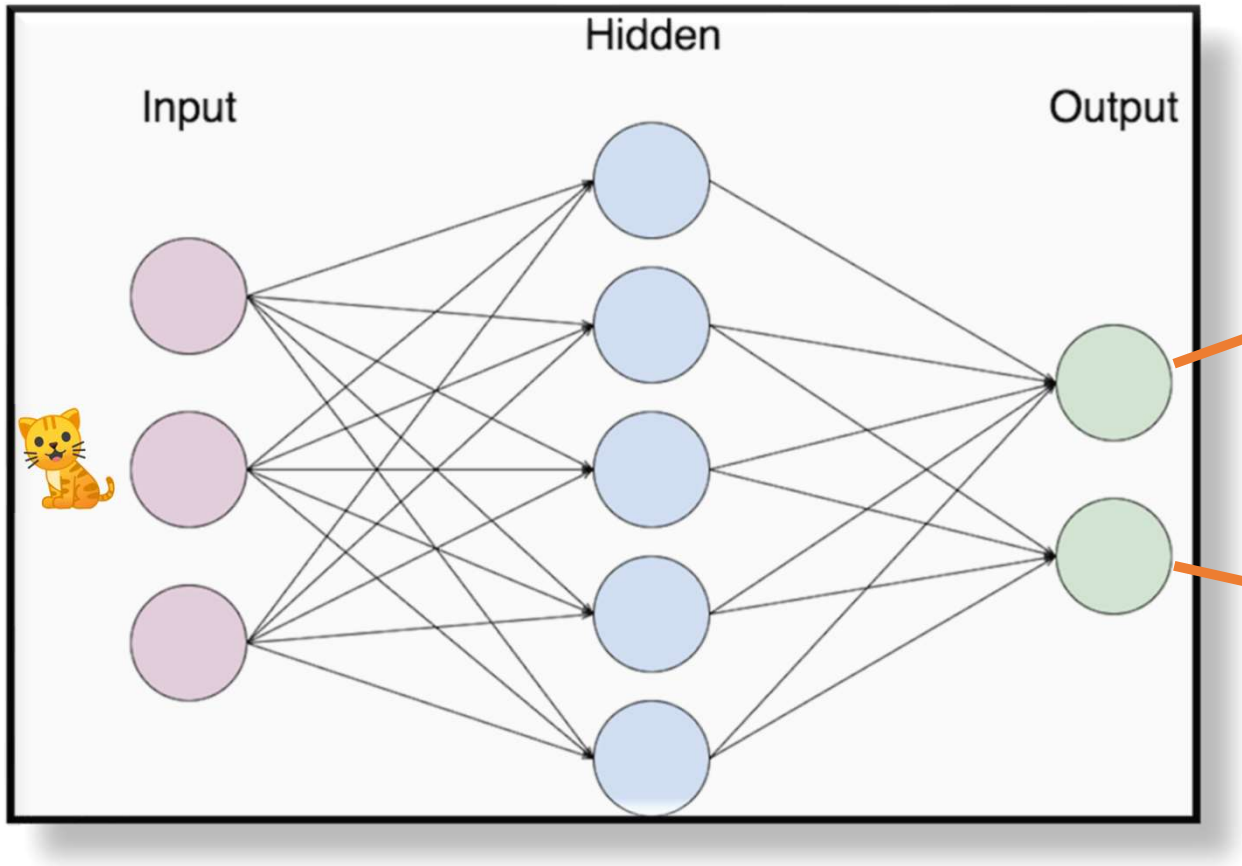
Multi Layer Perceptron



Neural Networks – Learning (Training)



ANN Learning – Minimizing Loss

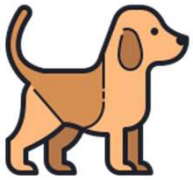


$$\frac{\partial loss}{\partial weight} * learning\ rate$$

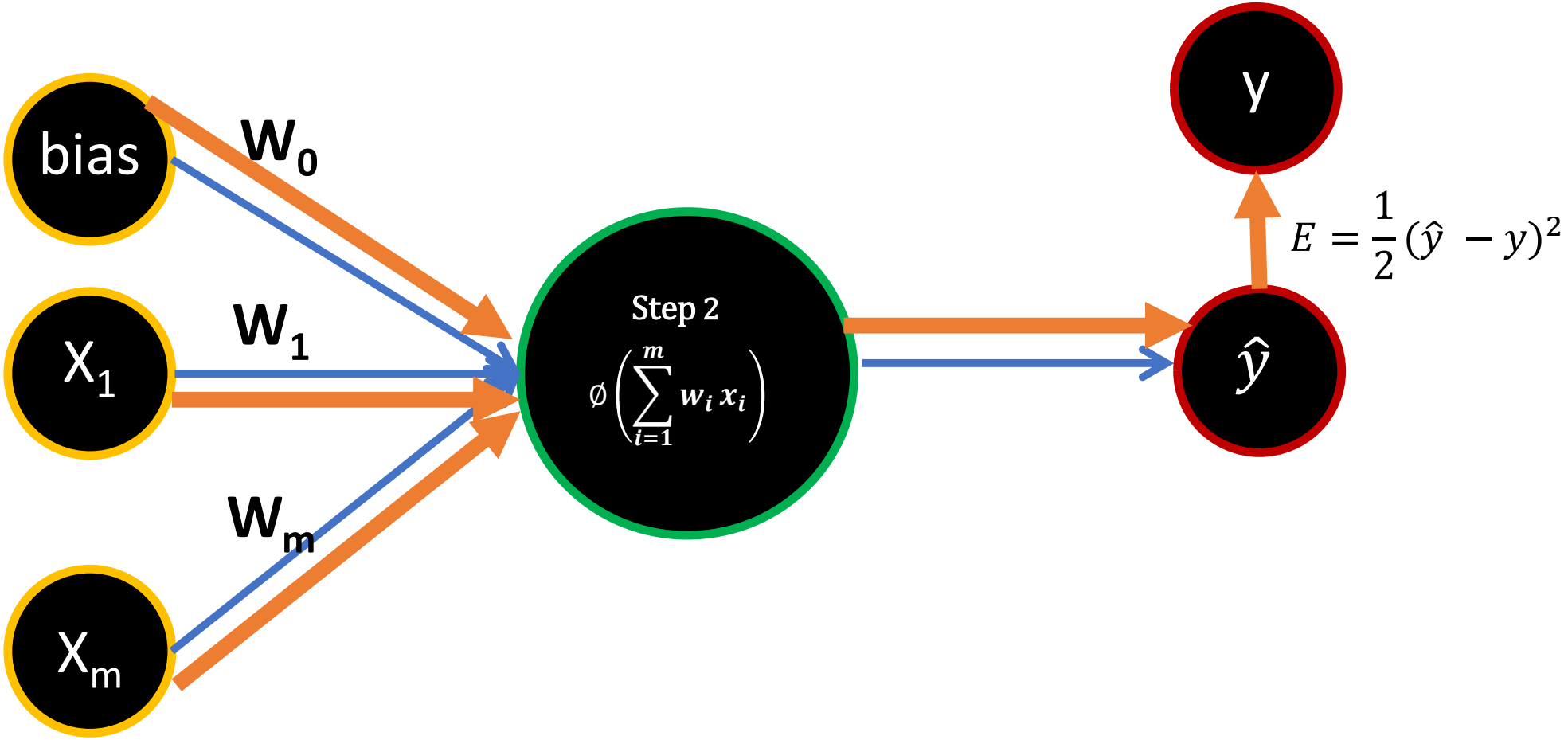
0.70

Range(0.01, 0.0001)

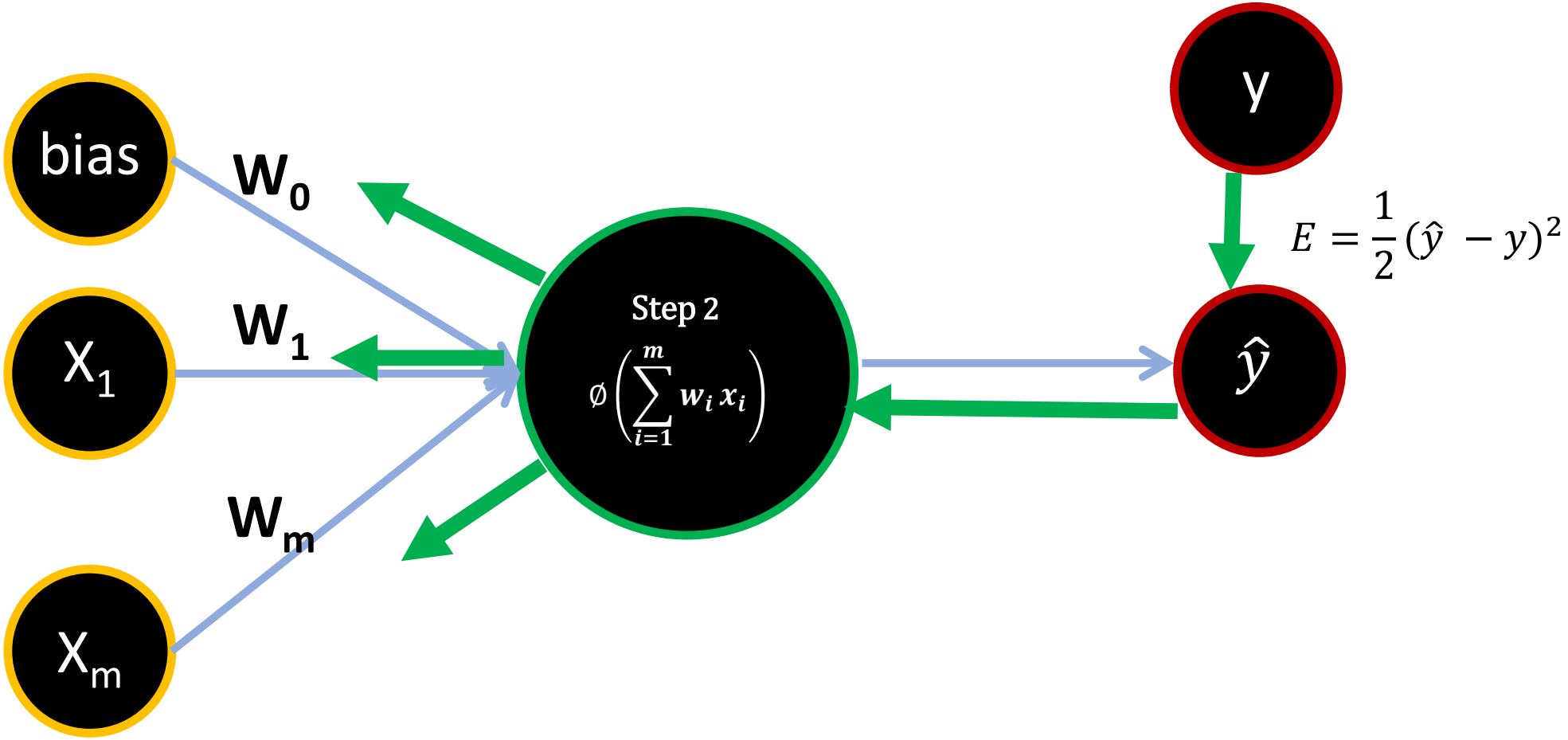
0.30



Neural Networks – Feed Forward

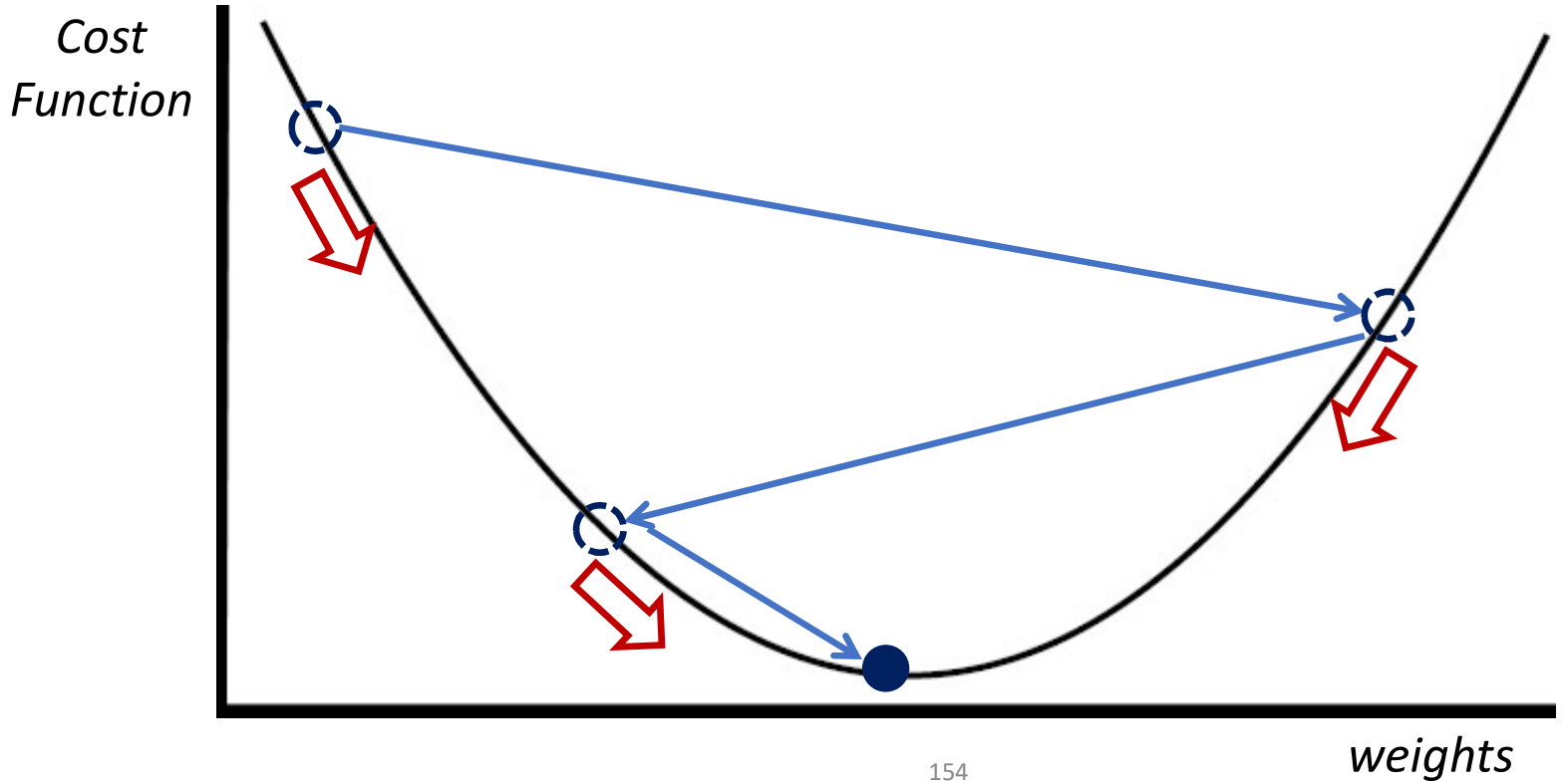


Neural Networks – Back Propagation



Gradient Descent

Used to learn the weights from the training dataset so that error can be minimized



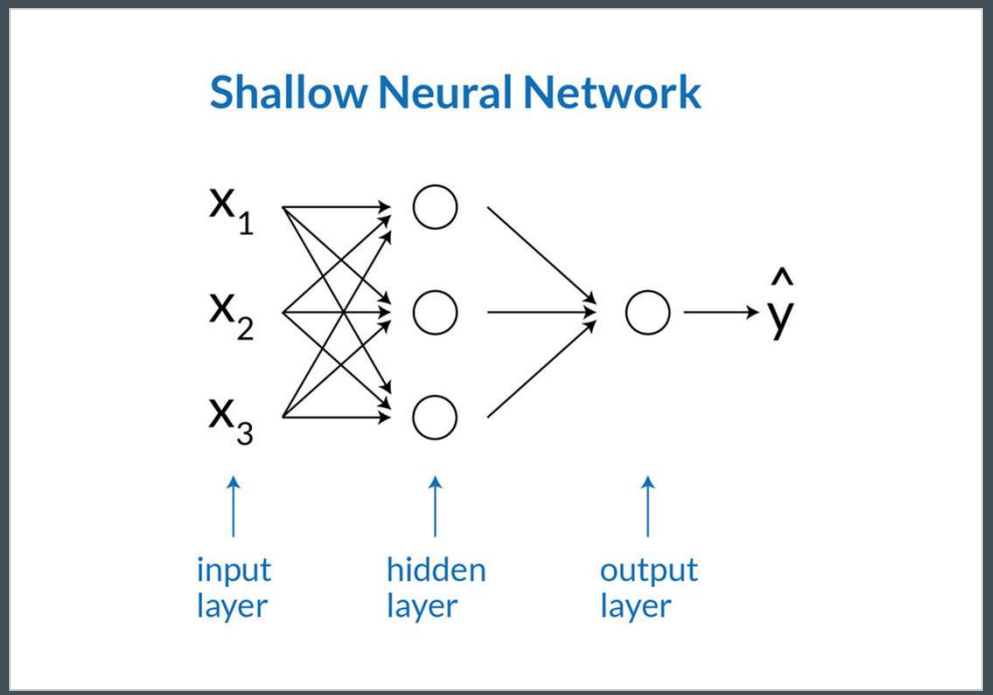
Neural Networks – Gradient Descent

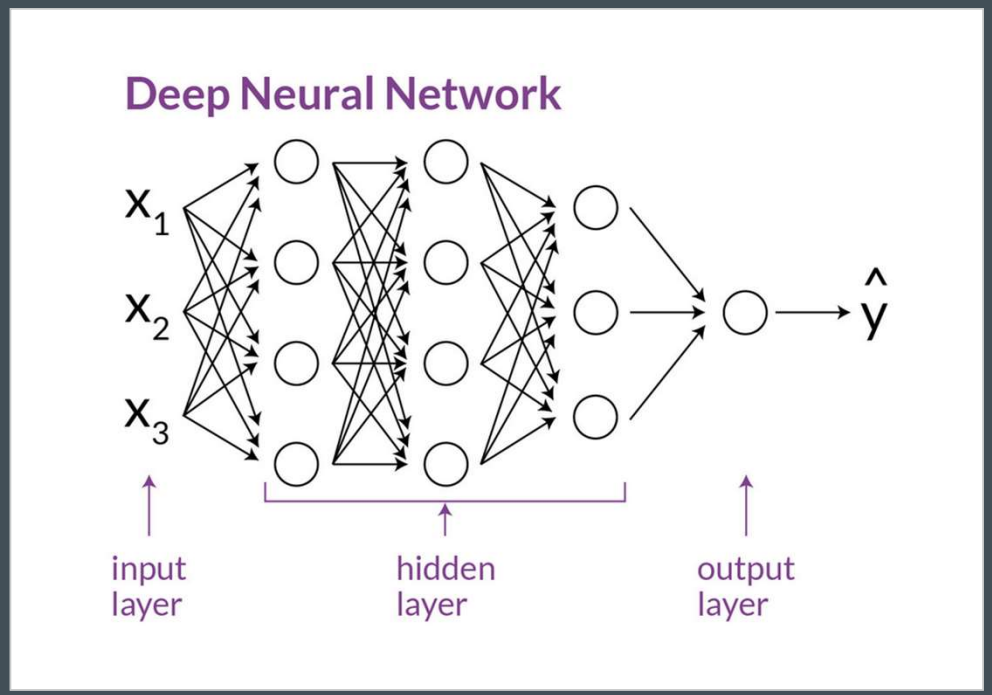
- Open file ‘CodeSamples/GradientDescent’ using Jupyter
- Examples of weight, loss, bias and epochs
- Plotting the values

TensorFlow, Keras – Basic Model

- Open file '**CodeSamples/TensorFlow-BasicModel**' using Jupyter
- Basic model to illustrate the use of TensorFlow, Keras
- View loss function and the optimizer

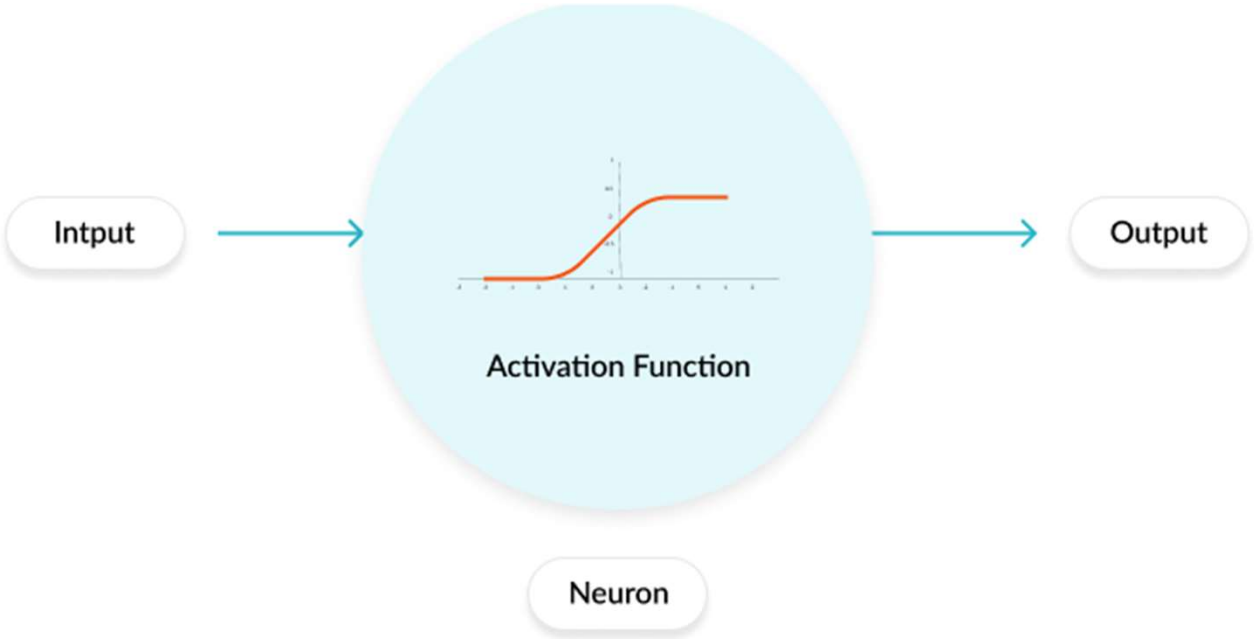
Shallow v/s Deep Neural Networks



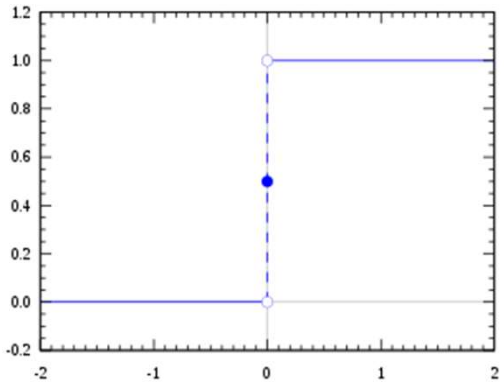


Activation Functions

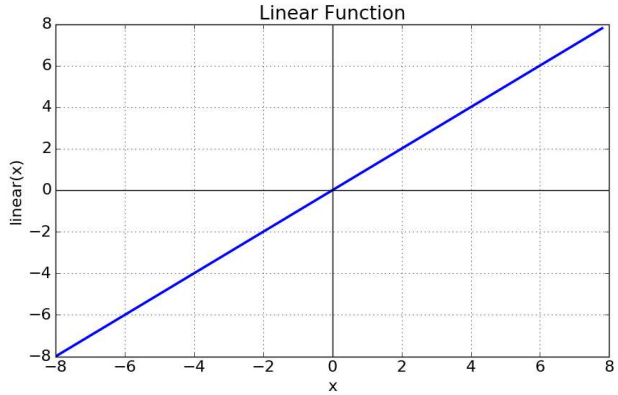
Activation Function



Step Functions

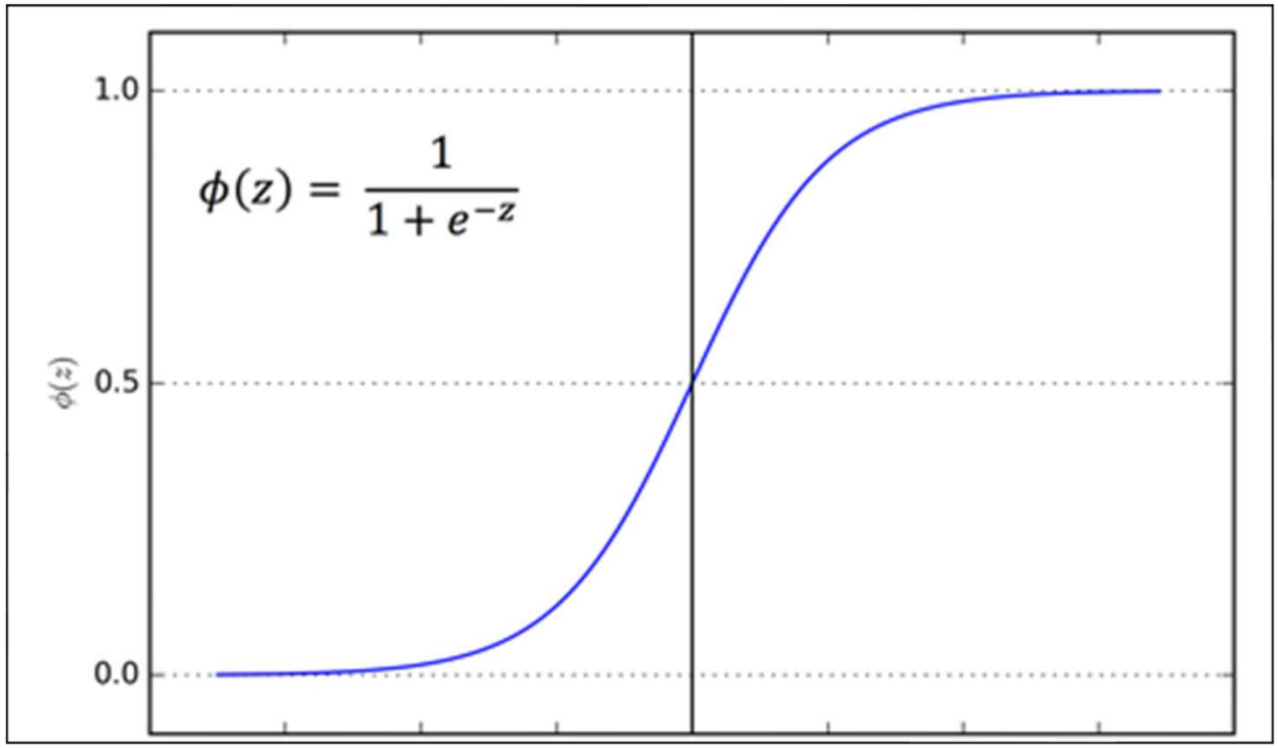


Binary Step Function

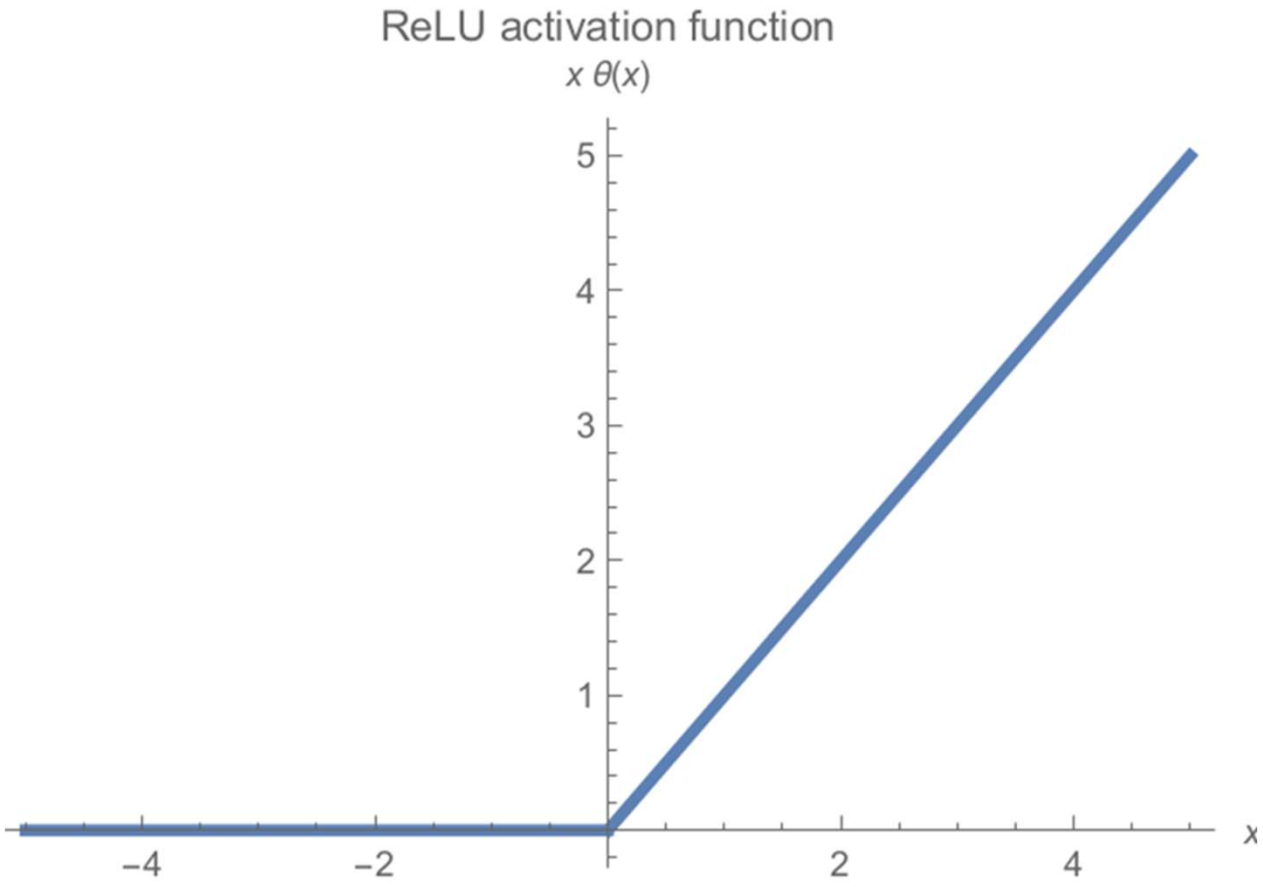


Linear Activation Step Function

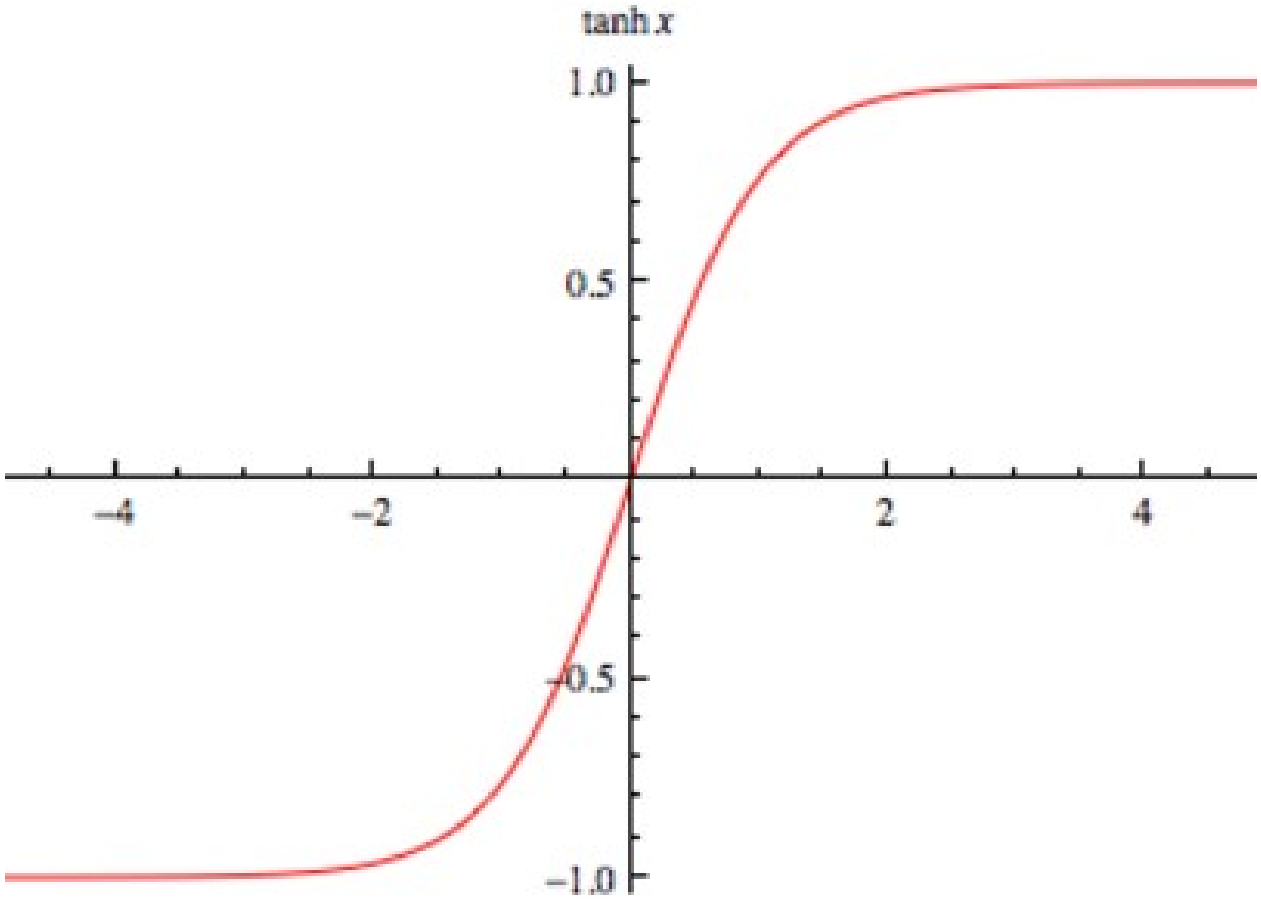
Sigmoid Function (Logistic)



Rectified Linear Unit (ReLU)



Hyperbolic Tangent (Tanh)

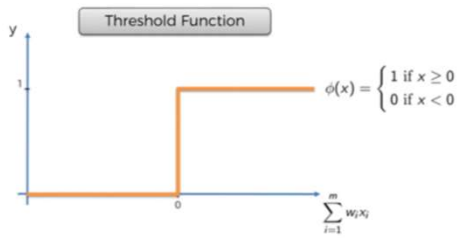


Softmax

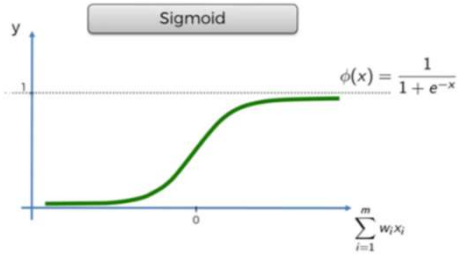
When handling multiple classes of output, Softmax function will give probability distribution for each class

Useful for finding class which has maximum probability

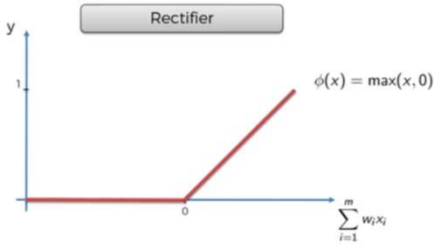
Often used in the output layer of multi-class classification



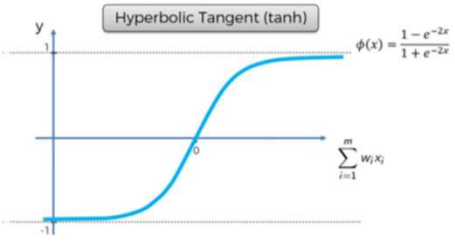
Binary Output Layer



Binary Output Layer



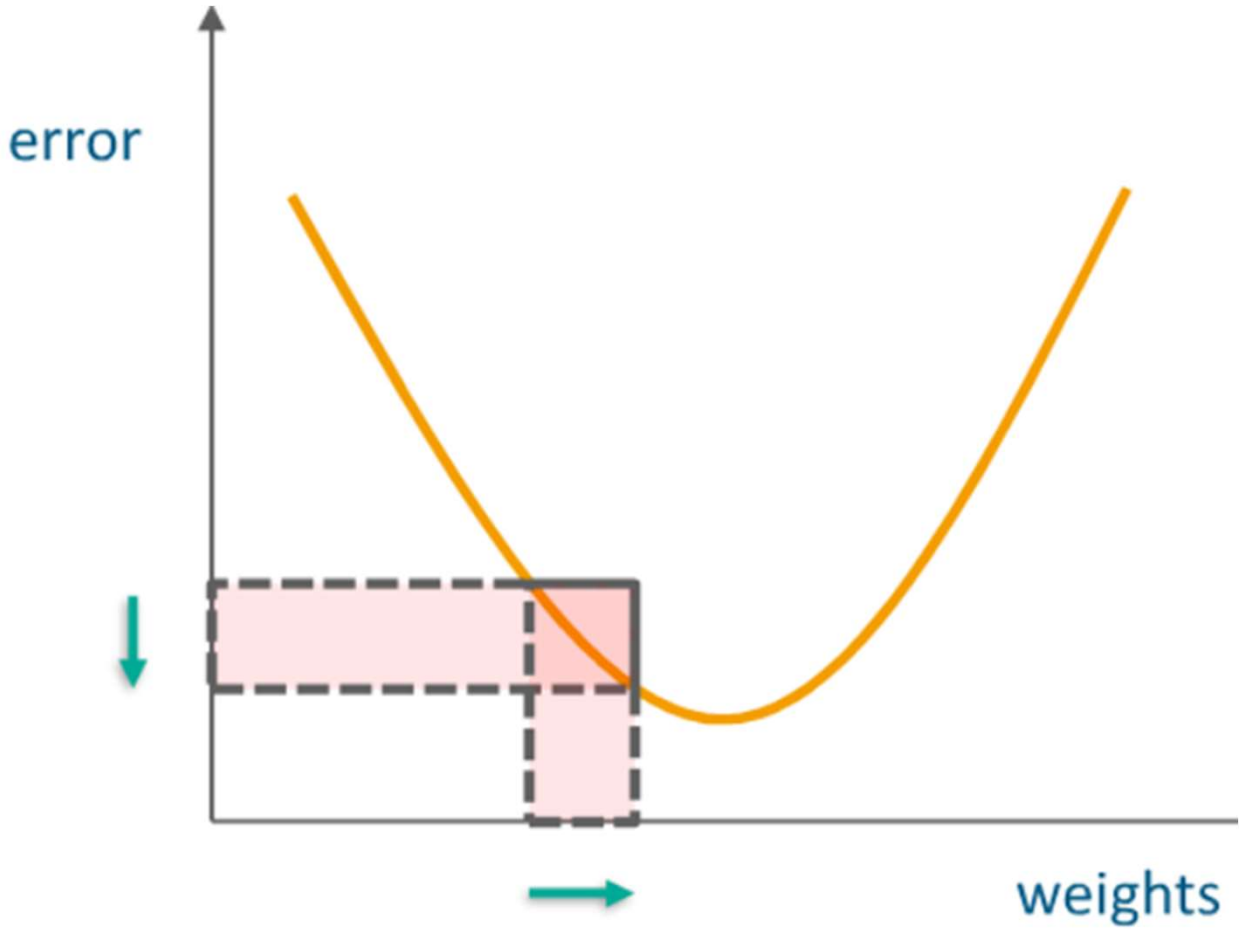
Hidden Layer: CNN



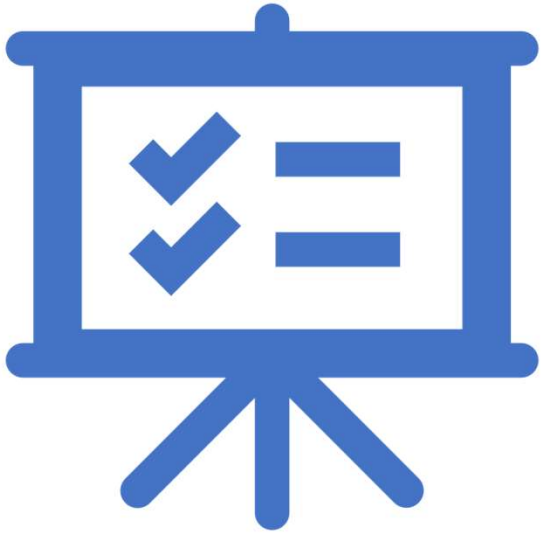
Hidden Layer: RNN

Neural Networks – Activation Functions

Learning Rate



Extra Credits



Extra Credits

- There are 4 datasets:
  1. Credit Card Application
  2. Churn Modelling
  3. Heart Disease
  4. Breast Cancer
- Your Work
  - Pick anyone of these datasets
  - Apply the techniques that you have learnt related to Feature Cleaning, Feature selection, etc.
  - Choose 1 or more algorithms and make predictions
  - Tomorrow afternoon we will review it with the class
- Description of datasets in following slides

Financial Customer Churn Data

- Dataset – Customer information with a financial institution
- Can use any algorithm to do classification (binary)
- Dataset has following features:

Dataset - Churn_Modelling.csv

RowNumber: Dataset row number
CustomerId: Customer Id
Surname: Last name of the person
CreditScore: Credit Score of the person
Geography: Country of residence
Gender: Person's Gender
AGE: Age of the person
Tenure: How long has the person owned the card
Balance: Outstanding balance
NumOfProducts: Number of products owned by the person with company
HasCrCard: Person has credit card
IsActiveMember: Is the person active member of the company
EstimatedSalary: Estimated salary of the person
Exited: Did the person stay or leave

Credit Card Application

- Based on 15 features (not named), predict whether a new customer's credit card application should be approved
- Predicting a "class" – 1 or 0

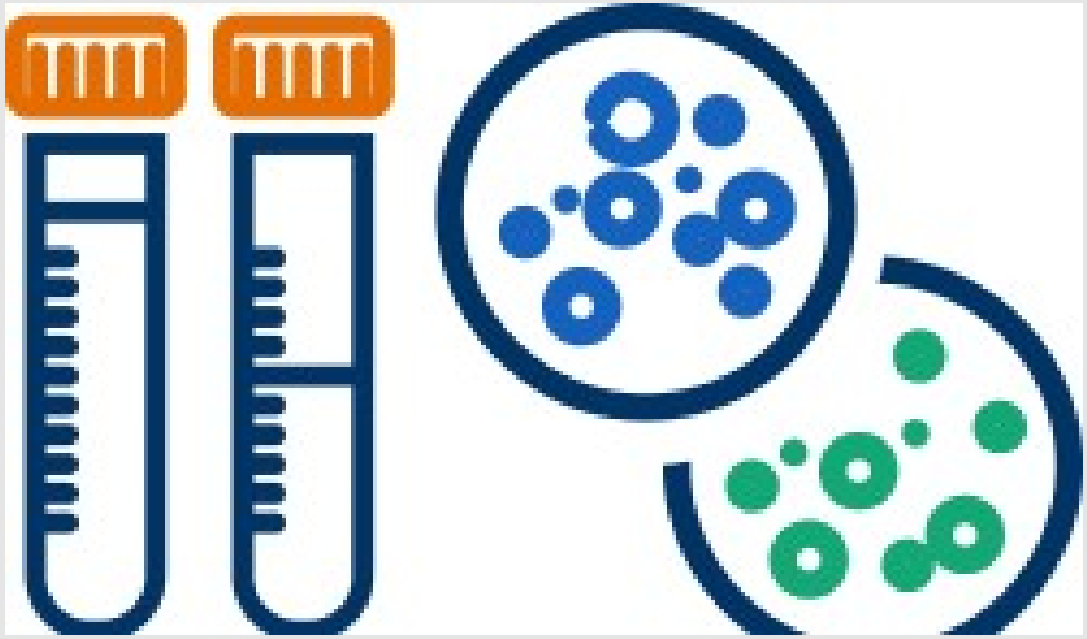
Dataset – Credit_Card_Applications.csv

Heart Disease

Dataset – Heart.csv

- Based on 13 features predict if a patient will have heart disease
- Features include age, gender, chest pain, resting blood pressure, cholesterol, fasting blood sugar, resting ECG, max heart rate, exercise induced angina, etc.
- Predicting a “target” – 1 or 0

Breast Cancer



- Dataset includes a number of features related to tissue samples collected
- 2 Classes: benign and malignant (values 2 and 4 respectively)
- Attributes:
  - Sample code number
  - Clump Thickness,
  - Uniformity of Cell Size,
  - Uniformity of Cell Shape,
  - Marginal Adhesion,
  - Single Epithelial Cell Size,
  - Bare Nuclei,
  - Bland Chromatin,
  - Normal Nucleoli
  - Mitoses

Dataset:

breast-cancer-wisconsin.csv

Neural Networks - Visual

- Use the link provided in the chat session to understand neural networks
- Tinker with number of hidden layers, number of neurons in each hidden layer, activation functions, etc.

Neural Networks - Regression

- Open file '**CodeSamples/NeuralNetworks-Regression**' using Jupyter
- Uses Auto MPG dataset from the UCI repository
- Predict fuel efficiency of late-1970s and early 1980s car models
- Dataset includes Cylinders, Displacement, Horsepower, Weight, Acceleration, Model Year, Country (US, Japan, Europe)

Assignment

- Open file '**Assignment/CRM-CustomerChurn**' using Jupyter
- Implementation requirements are defined in the notebook