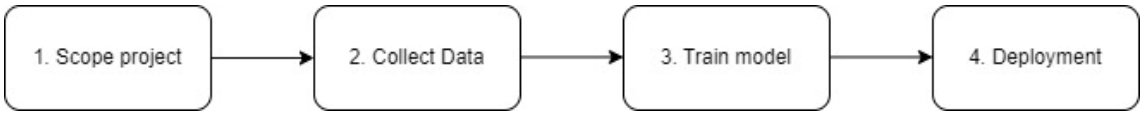
MACHINE LEARNING FRAMEWORK FOR SUPERVISED LEARNING APPLICATIONS

INDEX

0. MACHINE LEARNING DEVELOPMENT CYCLE

The full cycle of machine learning development is composed of 4 steps. First, the project has to be defined and scoped. Second, the developers must get data to train their model. Third, the model will go through the "Training iterative loop" where it will be trained and improved. Finally, the model will be deployed.

Basic Graphical ML – Development cycle



1. SCOPE PROJECT

1.1. Define project/problem

In the world are many possibilities and a lot of problems. All the applications are different, with different needs and requirements.

What problem do you want to solve? What are your inputs and what are your outputs?

The problems or project you want to do can be defined looking at the output. What is our desired output? You want to predict a number; you want to classify images?

Before you step out to the next phase of the development, be sure that the scope is well defined.

The **machine learning projects can be divided in**:

I. Predict an output (infinite possibilities)
II. Classify data
III. Look for any structure in data

1.2. Choose theorical model

Now we have to choose some **algorithm for the model**, this guideline is supposed for people that knows the basics algorithms and their uses

The machine learning projects can be divided in:

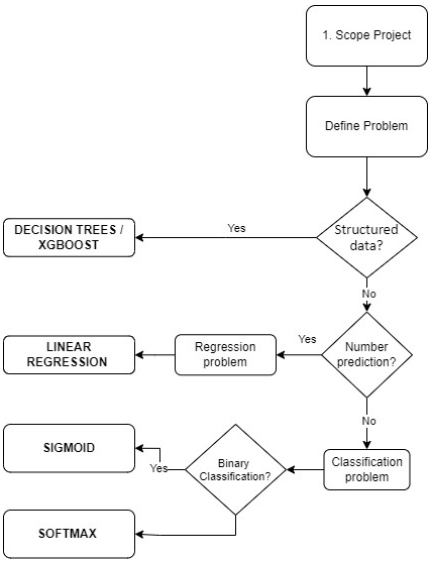
For:

Decision trees

For unstructured data:

I. Predict a number→ **Regression**
II. Classify data→ **Classification**
    a. Binary classification →**Sigmoid function**
    b. Non-binary classification → **SoftMax**
III. Look for any structure in data → **Unsupervised learning**

Graphical ML – SCOPE PROJECT



2. COLLECT DATA

2.1. How to get data?

The data may be the **most expensive thing** to get in Machine Learning projects, so getting good quality data with the lowest cost must be a priority.

There are **sites** to get free data in order to make your projects:

a. Data.gov
b. Google Dataset Search
c. Kaggle
d. Open Data Network
e. Data.gov
f. DataHub.io
g. Many more on the Internet……

This is a way to get data for free. But there are some cases that you want particular data for a company of the customers, in that case you would like to do things like **surveys**.

2.2. Data augmentation / Data synthesis

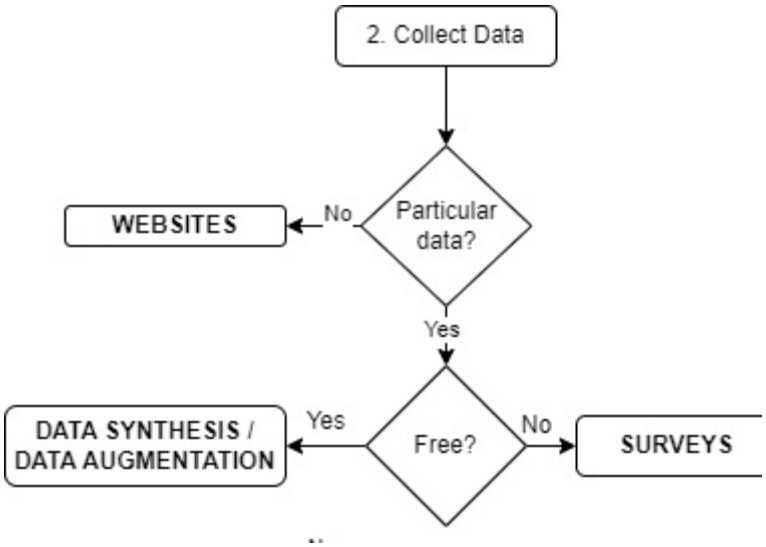
These two techniques are useful when you want more data and you have data stored in your database that you have used to your model.

The **Data augmentation** technique is based in modifying the data you have to create new. E.g., in speech recognition: add noise to your audios; In face recognition: rotate the photo a little.

The **Data synthesis** technique is based on creating new data for yourself. E.g., in speech recognition: create new audios

These two techniques can have a very positive impact in the training.

Graphical ML – COLLECT DATA



3. TRAIN THE MODEL

3.1. The training iterative loop

In this step is the meat and potatoes of the ML development process. This specific step of the full process has its own iterative loop. The tree steps are:

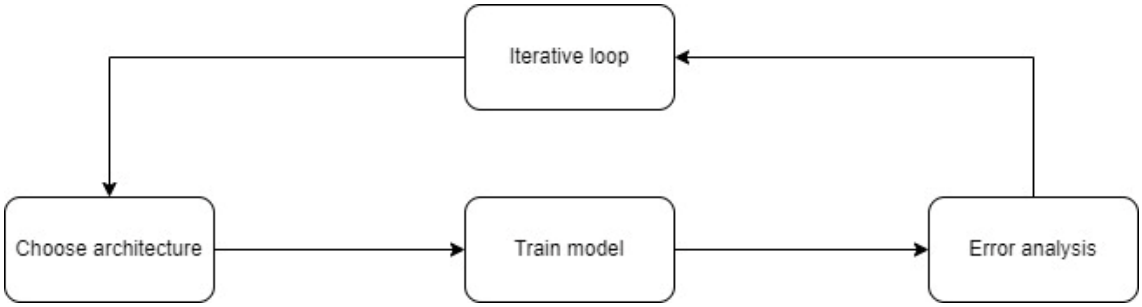
1. Choose architecture

2. Train the model

3. Error analysis

We will explain the process:

Graphical ML – TRAINING ITERATIVE LOOP



3.2 Choosing the architecture

In this framework we will use neural networks to perform our models. The neural network will start very simple and it will be improving while is going through all the iterations.

3.2.1 Base architecture

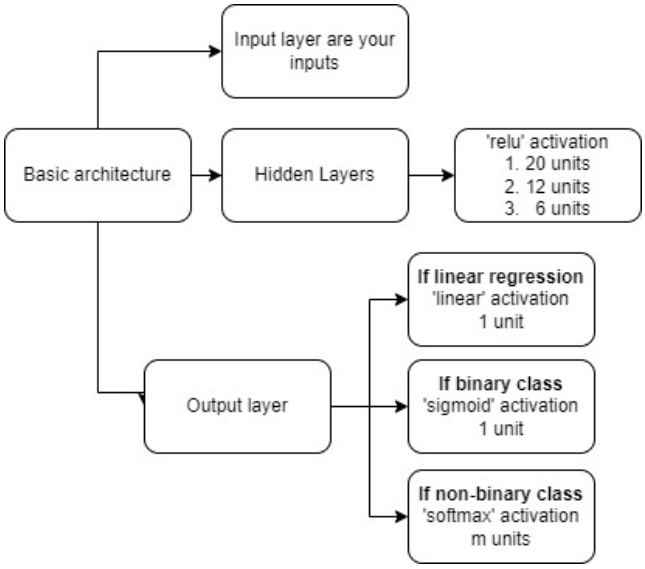
This architecture will only be used on the first iteration of the ML development.

The **input layer** will be your data. Your inputs.

The **hidden layer** at the beginning will start with: 3 layers of 20,12,6 units respectively. With 'relu' activation function.

The **output layer** changes depending on your output. If is a regression is one unit with the 'linear' activation function. If is a binary classification is one unit with the 'sigmoid' activation function. If is a non-binary classification is m units (being m the number of possible classes your model will predict) and activation function 'SoftMax'

Graphical ML – BASE ARCHITECTURE



3.2.2 Architecture improvement

We have seen the base architecture for the first iteration on the training process. But after one iteration, what we can improve in the architecture. The improve will be:

All the hidden layers multiply by 2 the units. Add one layer with 3 times more units than the layer before.

Ex:

If we have the base architecture, the next iteration the architecture would be 60 -> 40 -> 24 -> m

And the next:

180 -> 120 -> 80 -> 48 -> m

3.2.3 Divide the dataset

The dataset must be divided in three parts:

1. 60% goes to training set →With this the model will be trained

2. 20% goes to cross validation (CV) set →With this, the model is tested before the final test

3. 20% goes to the test set → With this, if the model is proved with cross validation set, it is tested like a 'final exam' whit the test set.

3.3 ERROR ANALYSIS

After training the model with the training set is very rare that we get the model at first try. So, how we can improve the model?

3.3.1 How the error is calculated?

As a simple definition, the error is the difference between the correct answer and the answer that our model has generated.

For **Regression → Mean Squared Error**

$$J_{train}(\vec{w}, b) = \frac{1}{2m_{train}} \left[\sum_{i=1}^{m_{train}} (f_{\vec{w},b}(\vec{x}^{(i)}_{train}) - y^{(i)}_{train})^2 \right]$$

$$J_{cv}(\vec{w}, b) = \frac{1}{2m_{cv}} \left[\sum_{i=1}^{m_{cv}} (f_{\vec{w},b}(\vec{x}^{(i)}_{cv}) - y^{(i)}_{cv})^2 \right]$$

For **Classification → % of misclassifications**

$$Error = \frac{n^{\circ}\ of\ missclassifications}{m}$$

NOTE: The error notation will be J(train) for training set, J(CV) for CV

3.2.2 Bias, Variance and Baseline Performance

This guideline is done for who that understand bias and variance.

The neural networks or models can **have three types of problems.**

1. High bias

2. High variance

3. High bias and high variance

Every problem has different solutions.

The baseline performance is the base error that will be compared to our model.

Types of baseline performance (BP) are:

1. Human error

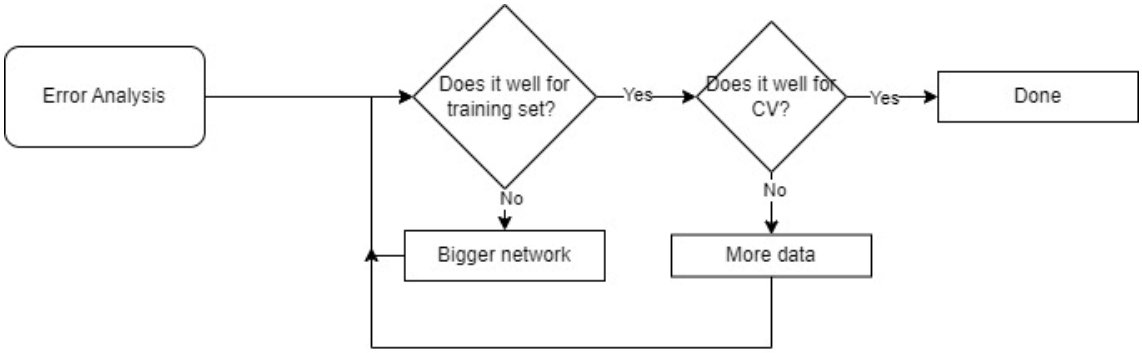
2. Guess by experience

3. Compare to other algorithms at the same task

CALCULUS	ERROR	SOLUTION
BP << J(train)	High Bias	**Add polynomials (x1^2, x2^2.)** **Increase neural network** Increase regularization (lambda) More features, feature engineering
J(train) << J(CV)	High Variance	**More training examples** **Increase neural network** Decrease regularization (lambda) Reduce features
BP << J(train) << J(CV)	High Variance, High Bias	EXIT, RETHINK

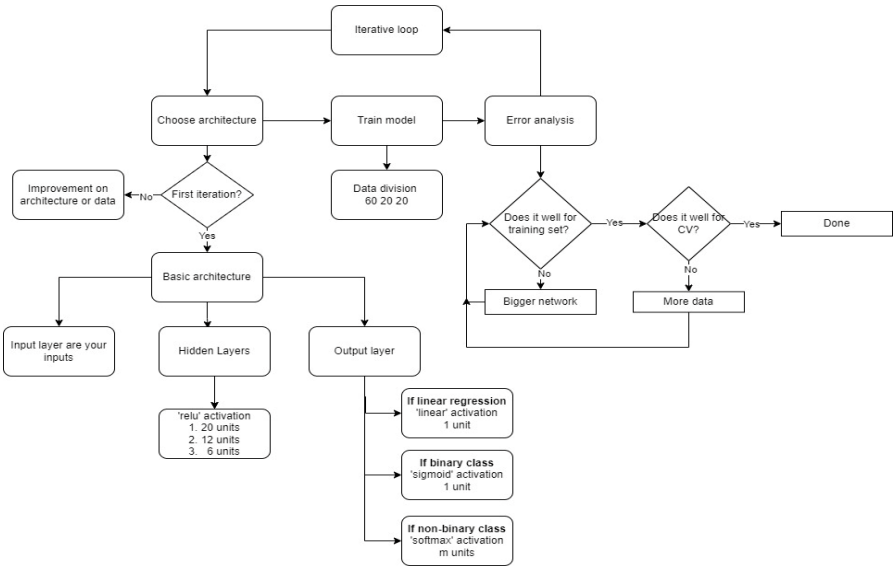
NOTE: << much smaller (relatively)

Graphical ML – RECIPE FOR NEURAL NETWORK



3.4 What to do next?

Now, we have detected all the errors and next steps to do:



3.5 Transfer learning

Using data from another task (not your actual problem / project) to train you neural network.

It works because is the same type of problem. For e.g. in a regression example, your neurons are trained for some problem that you have millions of data, and for your actual problem, you only have 100. If you train your layers with the other data, it will work because the neurons are working in the same problem.

Process:

1. Supervised pretraining

2. Copy all layers minus the output layer

3. Fine tunning -> train the trained model with your actual problem data

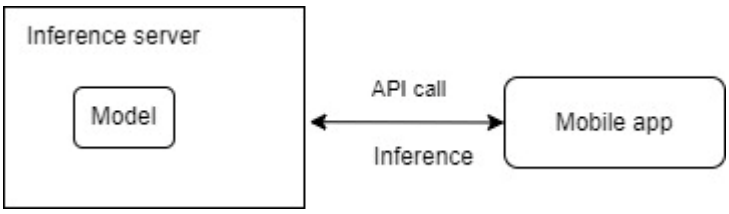
3.1 Only train the output layer

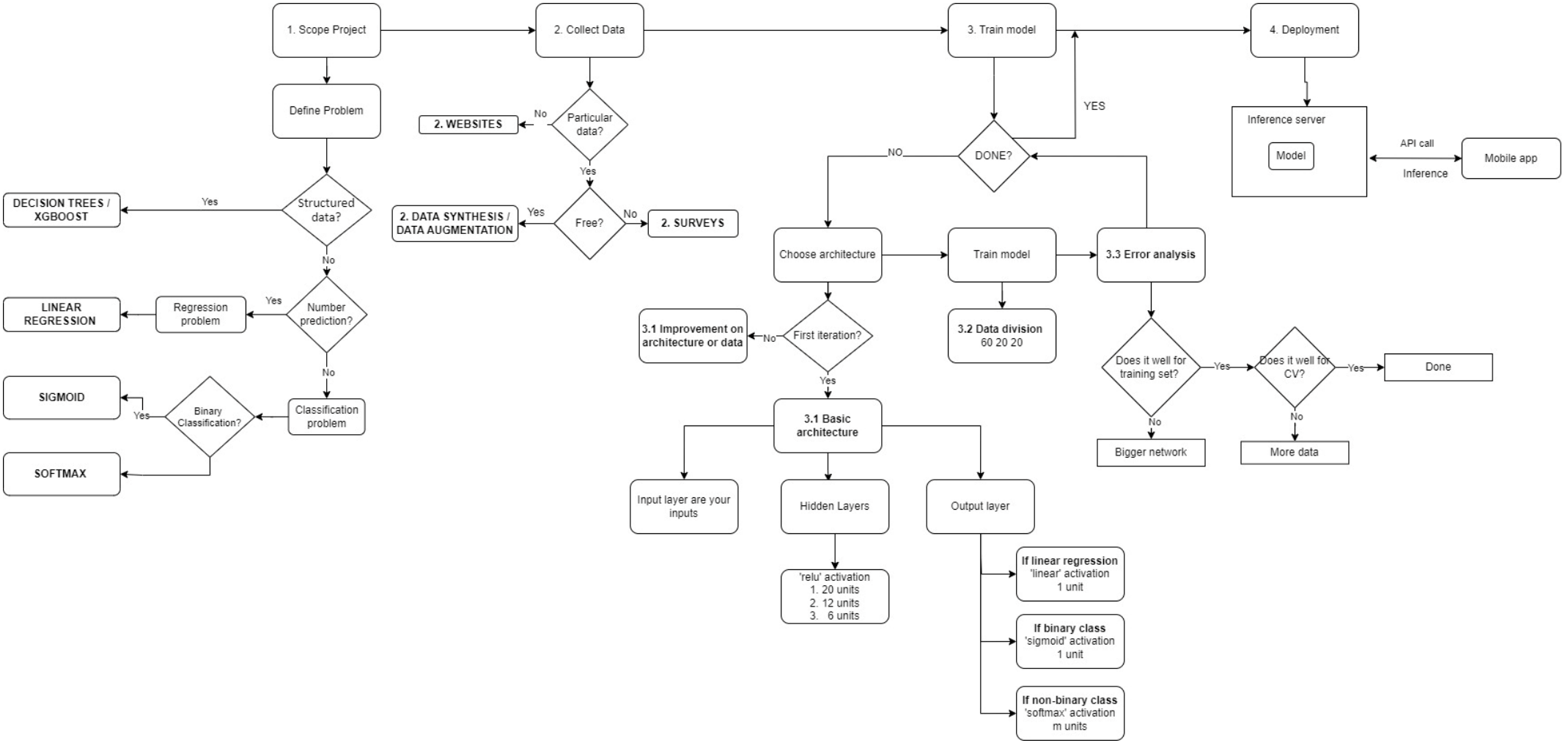
3.2 Train all the layers

4. Deployment

When you finalize your model, it has to be stored in a server.

Through applications like a mobile or web applications, you can do API calls to your model and respond those calls.




1. Scope Project
2. Collect Data
3. Train model
4. Deployment
Define Problem
Structured data?
Yes
DECISION TREES / XGBOOST
No
Number prediction?
Yes
Regression problem
LINEAR REGRESSION
No
Classification problem
Binary Classification?
Yes
SIGMOID
SOFTMAX
Particular data?
No
2. WEBSITES
Yes
Free?
Yes
2. DATA SYNTHESIS / DATA AUGMENTATION
No
2. SURVEYS
DONE?
NO
YES
Choose architecture
Train model
3.3 Error analysis
First iteration?
No
3.1 Improvement on architecture or data
Yes
3.1 Basic architecture
Input layer are your inputs
Hidden Layers
'relu' activation
1. 20 units
2. 12 units
3. 6 units
Output layer
If linear regression
'linear' activation
1 unit
If binary class
'sigmoid' activation
1 unit
If non-binary class
'softmax' activation
m units
3.2 Data division
60 20 20
Does it well for training set?
Yes
No
Bigger network
Does it well for CV?
Yes
No
More data
Done
Inference server
Model
API call
Inference
Mobile app