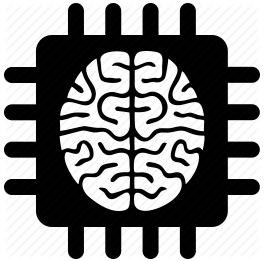
nDPI & Machine Learning



A future concrete idea

ntop

1. Conjunction between DPI & ML

2. Introduction to Tensorflow and ConvNet project

Traffic classification approaches

Category	Classification methodology	Attribute(s)	Granularity	Processing time
Port based	Protocol port	Protocol ports	High	Low
Payload inspection	Deep Packet Inspection	Payload inspection	High	High
	Stochastic Packet Inference	Statistical properties inherent in packet headers and payload	High	High
Behaviour Techniques	End-point monitoring	Identify host behaviour pattern	Low	Moderate
	Traffic accounting	Heuristic analysis of inspected packets, flows	High	High
Statistical Approaches	Packet based	Packet and payload size, inter-packet arrival time	High	Moderate
	Flow based	Duration, transmission rate, multiple flow features	Low	Low

The goal

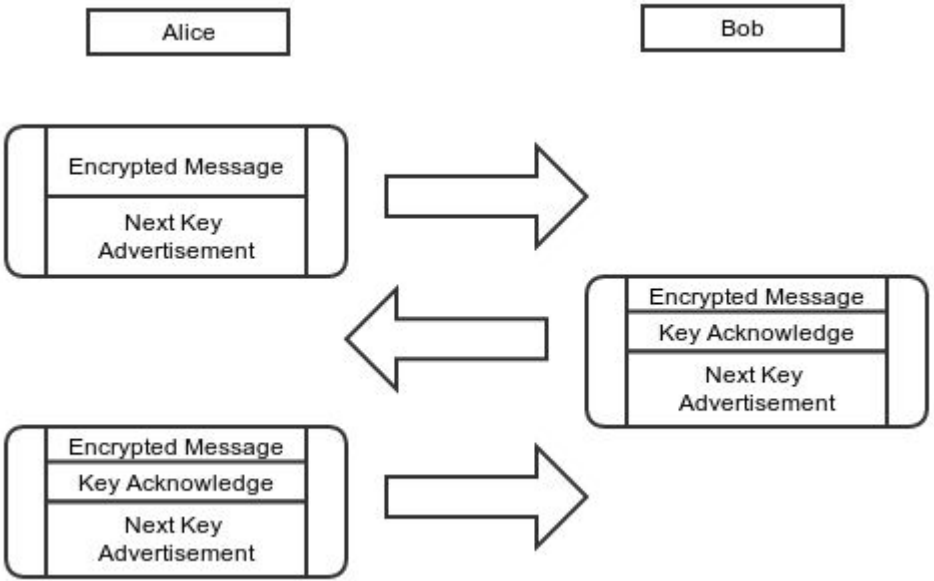
Try to find a union between the features of *DPI* and the potentialities of *ML* (Deep Learning)

nDPI

- Open-source library for protocol classification by the **payload**
- How a protocol is *detected:*
  - ***The concept of flow***
  - **Port matching -- Ip address matching**
  - **Dissector**
- Any example of detection:
  - Easy (i.e. *HTTP*)
  - Normal (i.e. *QUIC*, *RTMP*)
  - Hard (i.e. *TOR*, *Skype*, ***Whatsapp***)
  - Host / IP matched (i.e. *Facebook*, *Twitter*)

Case study: WhatsApp

Proprietary and encrypted:



Uses different protocols:

- **STUN** (Session Traversal Utilities for NAT) for NAT traversal of applications for real-time voice, video, messaging (usually on UDP)
- A customized version of **XMPP** (eXtensible Messaging and Presence Protocol) to handle the message delivery system
- **Signal** protocol to encrypt messages
- **SRTP** to encrypt WhatsApp calls

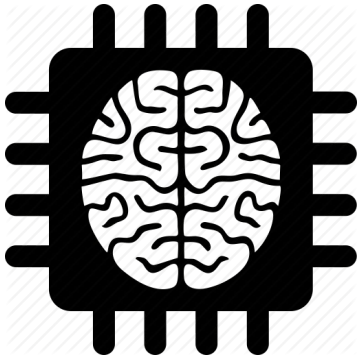
From DPI to ML 1/2

DPI libraries replaced the first generation of port-based tools, but

- Some countries forbid DPI
- DPI uses significant computing resources in order to be performed
- **DPI is limited by new implementations of a protocol (*dissectors must be updated every time there is a change*)**
- Encrypted payload

From DPI to ML 2/2

- Machine Learning is the technique that *"**gives computers the ability to learn without being explicitly programmed**"* (Arthur Samuel, 1959)
- Models:
  - Supervised
    - Dataset with label
  - Unsupervised
    - Dataset without label
- Dataset:
  - Hundreds thousand or millions pkts

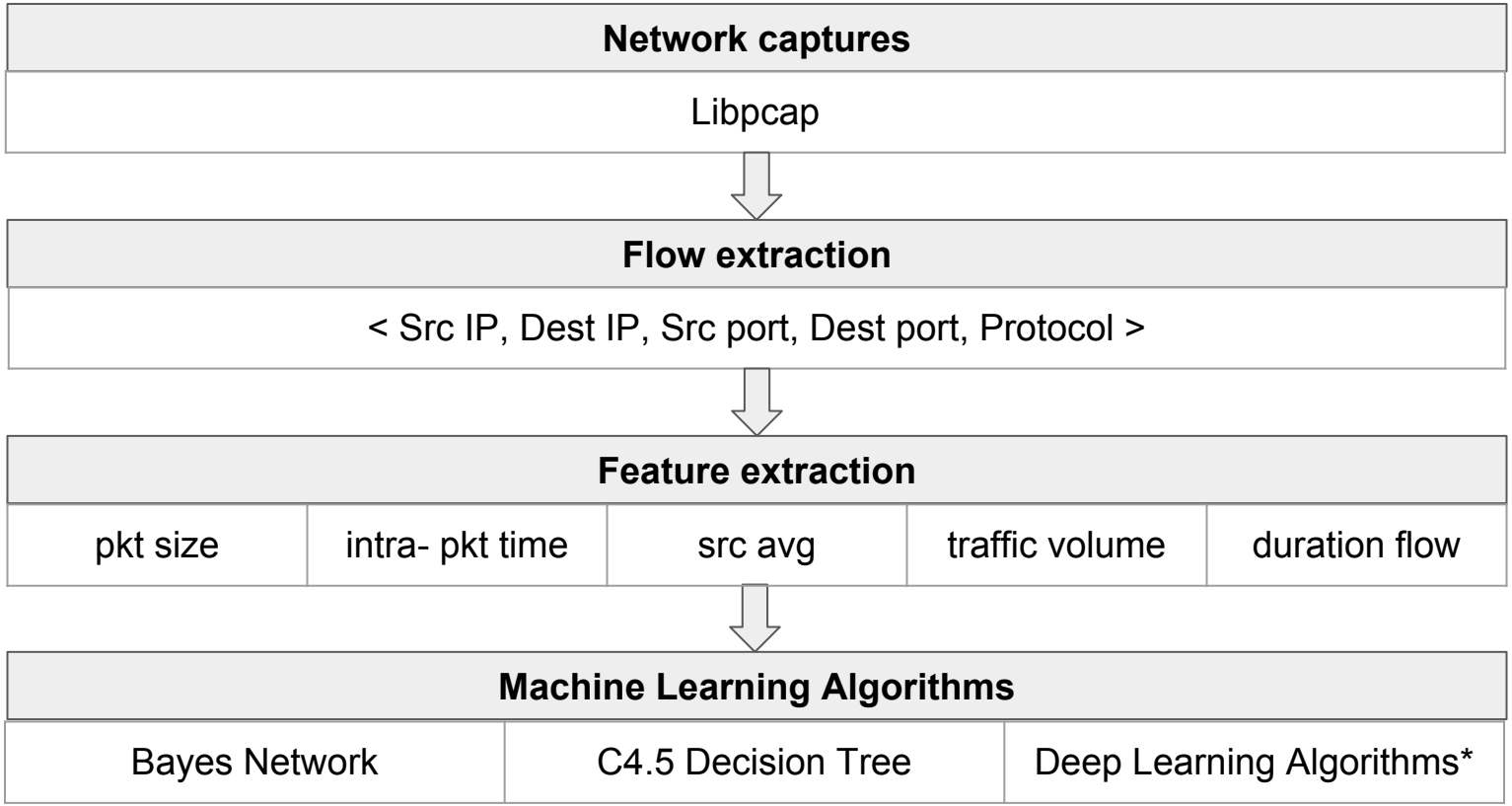


Machine Learning in protocol detection

Question: Can we extract features from a flow of WhatsApp to train a model for semi-automatic detection ?

Answer: This is exactly what we want to do with ML. Let's do it !

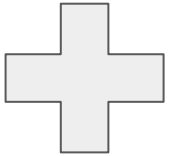
- **Flow** is a set of pkts where we extract statistical features and features values
- Flows are used to train the **classifier**
- Classifier is used to determine the **class** of flows



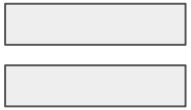
Process to extract features and train a model

**Deep learning has the advantage that we can avoid to define prior features*

- **Machine Learning to classify applications as WhatsApp, Facebook, etc.**

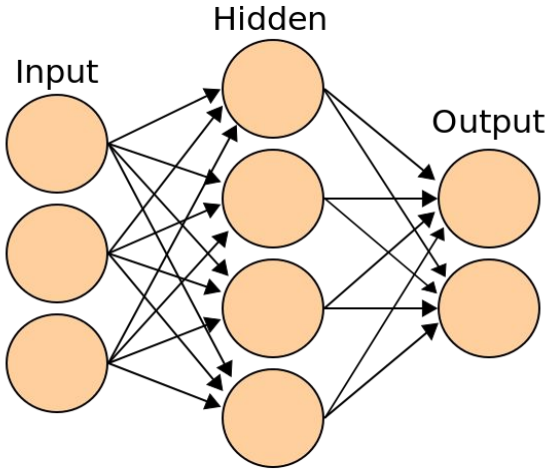


- **DPI to recognize protocol under the hood of applications**



Highest accuracy

A parallel work on Deep Learning

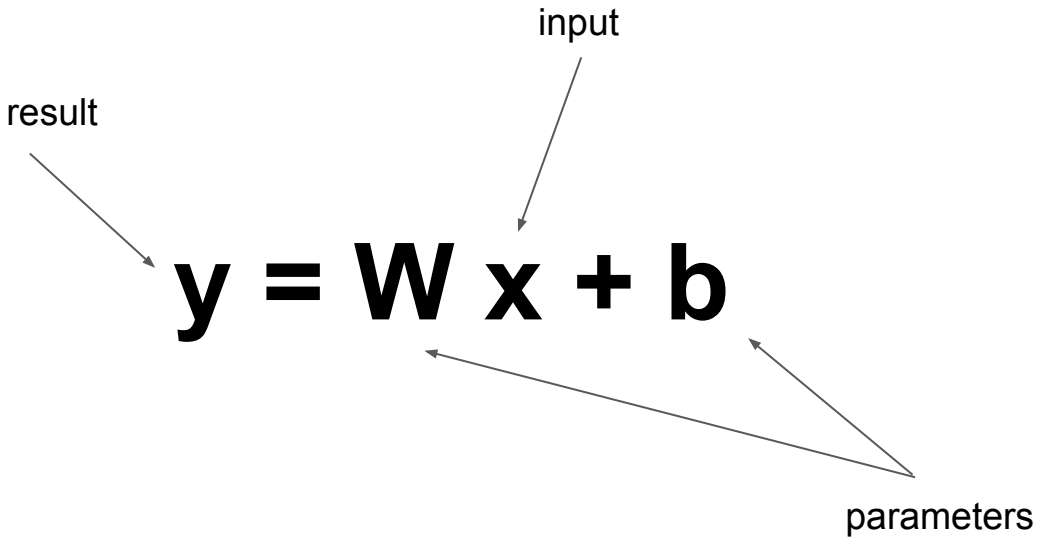


ConvNet

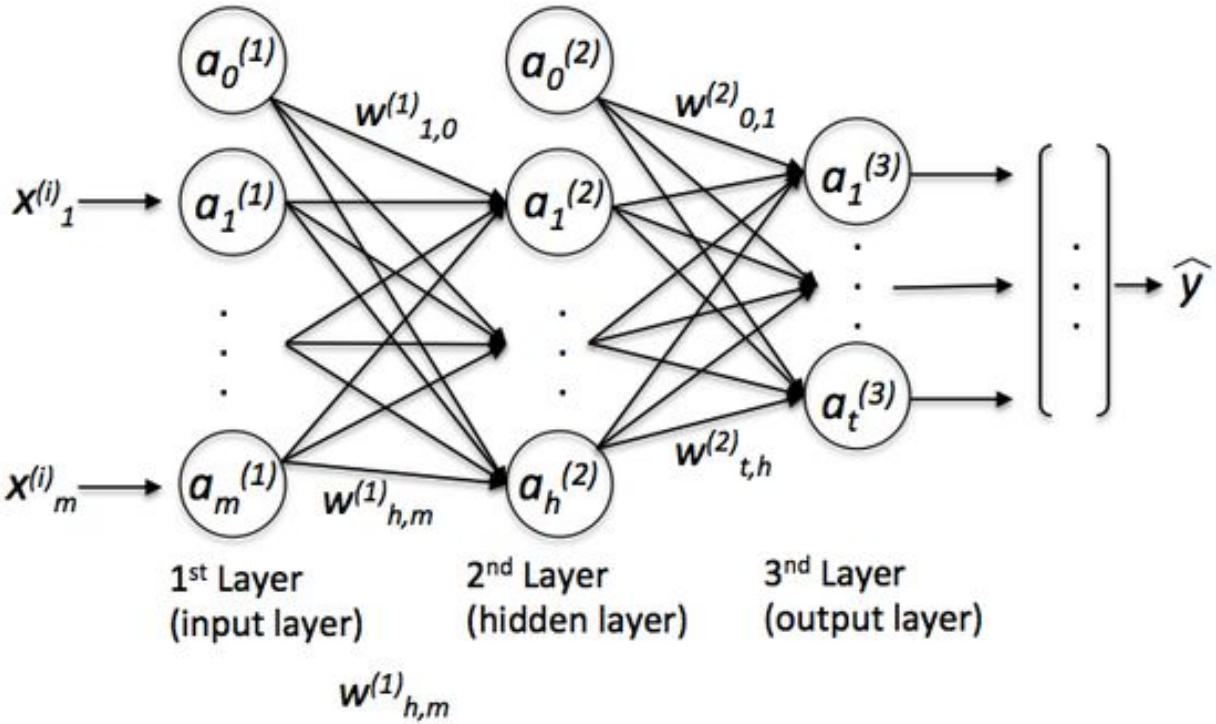
- **ConvNet** is a *convolutional neural network* for image recognition.
- Developed with **Tensorflow**
- Based on **AlexNet** algorithm
- Model build with several **convolutional layers**
- **Dataset** made by preprocessing images to create serialized object:
  - Load dataset at a reasonable speed (*Cpickle module*)
- Use **optimizer** Tensorflow function to minimize the ***loss*** (error on learn)
- **Model trained** with labeled images

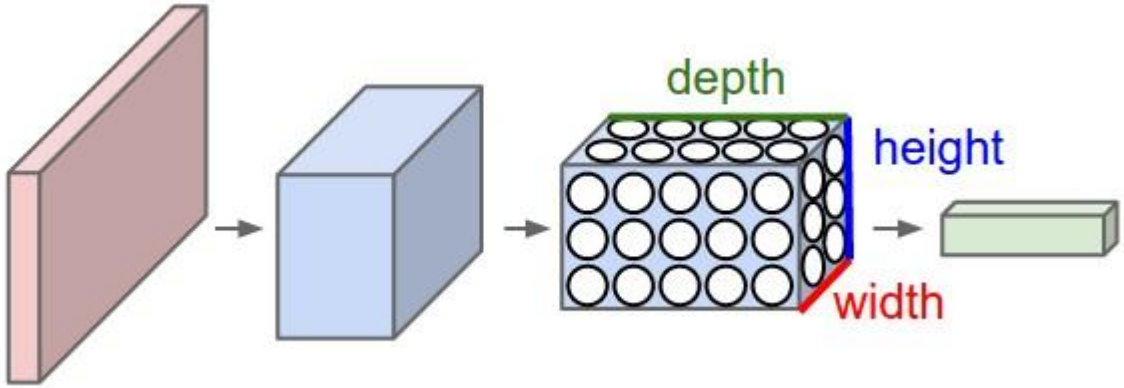
Fundamental concept:

- Linear Regression:
  - How to model the relationship between a scalar dependent variable **y** and one or more independent variables **x**

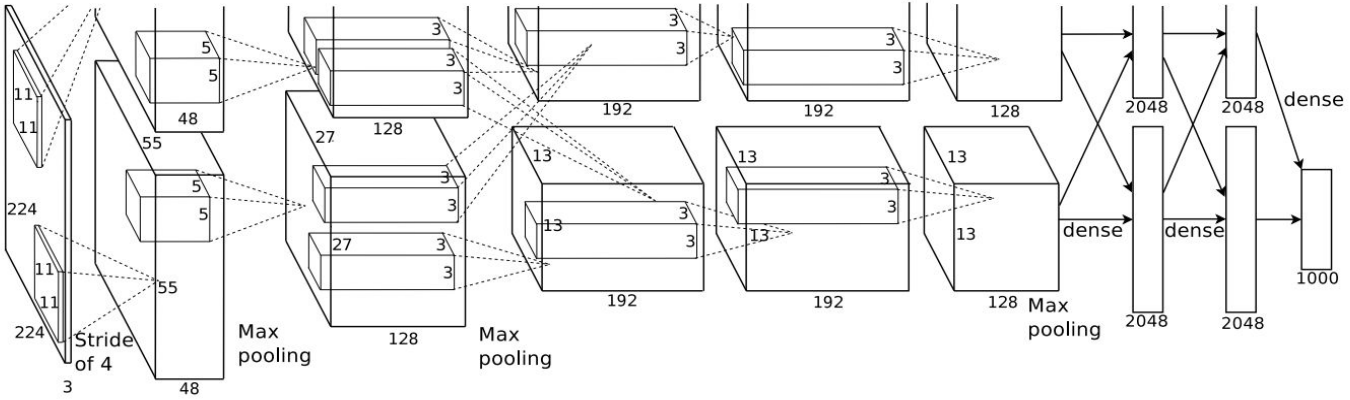


Logistic Linear Regression for a Deep Learning model





1. Convolution of an image



2. AlexNet base deep ConvNet

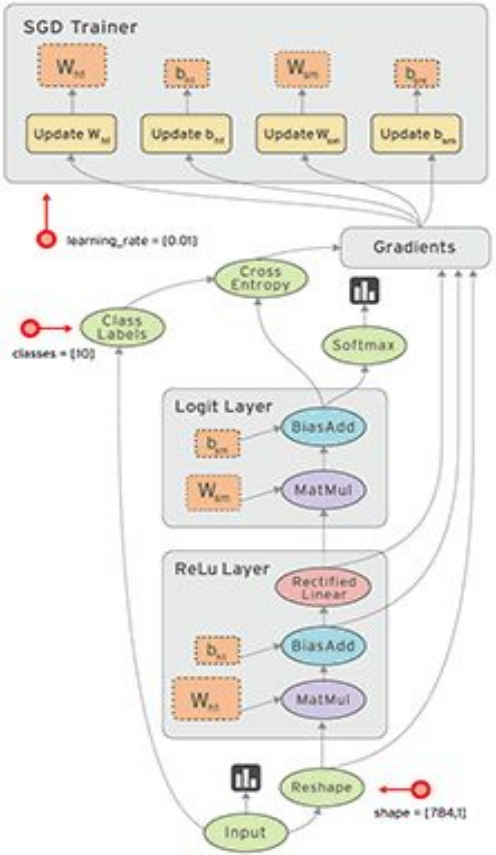
Tensorflow

Tensorflow is an open source library implemented by Google for machine learning using **data flow graphs:**

- **Computation** is defined as a graph
- **Nodes** in the graph represent mathematical operations
- **Graph edges** represent the multidimensional data arrays (**tensors**) communicated between them

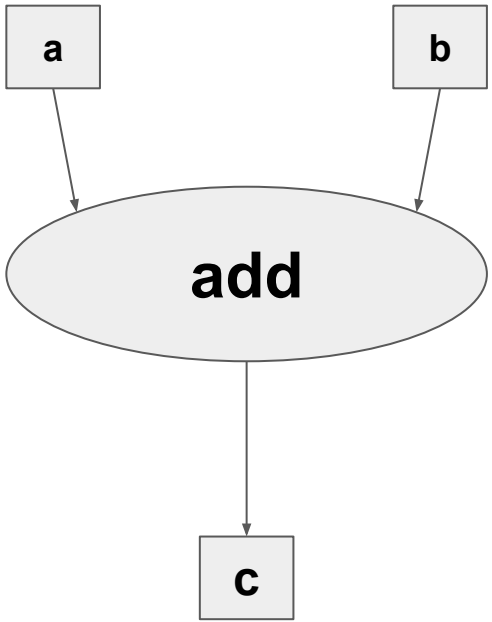
What is a Data Flow Graph?

- Data flow graphs describe mathematical computation with a graph of nodes and edges
- Graph is defined in **high-level language** (Python, C++)
- Graph is **compiled** and **optimized**
- Nodes represent computations and state
- Data (tensors) flow along the edges



Build a graph and then run it

```
import tensorflow as tf

. . .

c = tf.add( a,b )

session = tf.Session()

val_c = session.run( c, {a=2, b=3} )
```

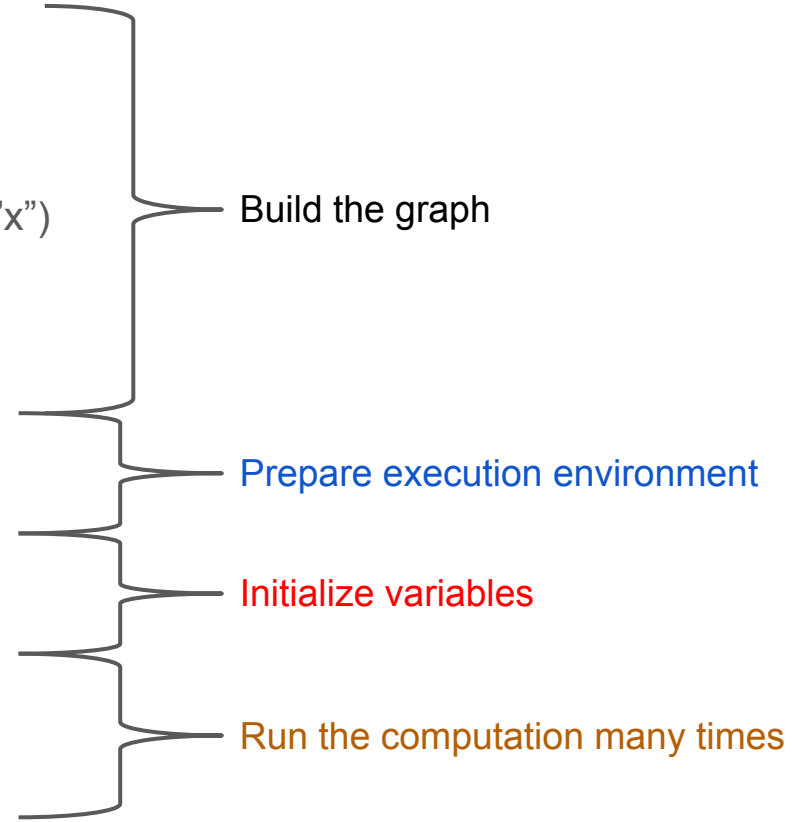


A session is the execution of a context

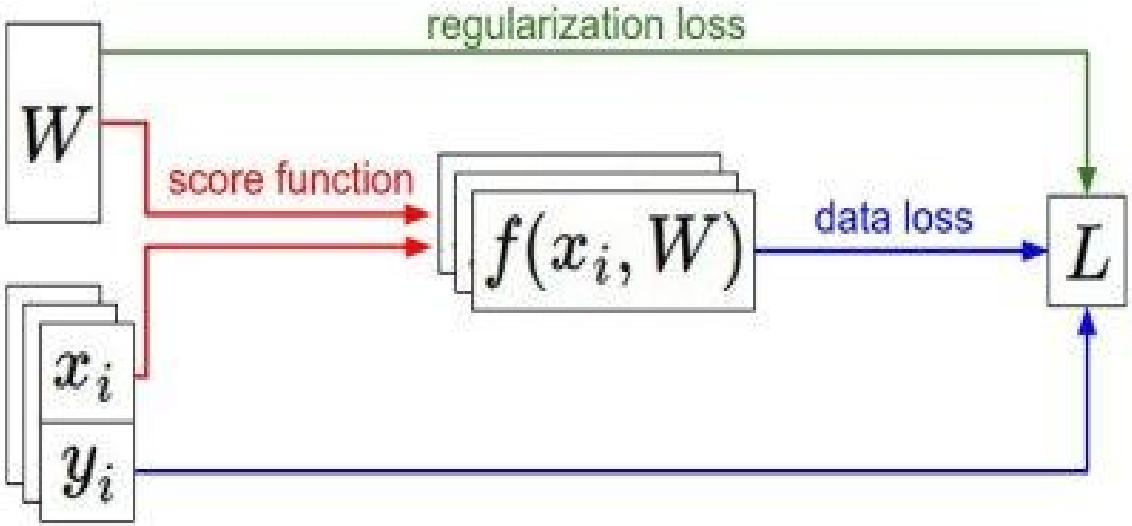
ntop

A simple implementation

```
import tesorflow as tf

x = tf.placeholder(shape = [None],
                                dtype = tf.float32, name=”x”)
W = tf.get_variable(shape = [], name=”W”)
b = tf.get_variable(shape = [], name=”b”)
y = W * x + b

with tf.Session() as sess:

    sess.run(tf.initialize_all_variable())

    For i in max_epochs:
        print(sess.run(y, feed_dict={x : x_inputs}))
```



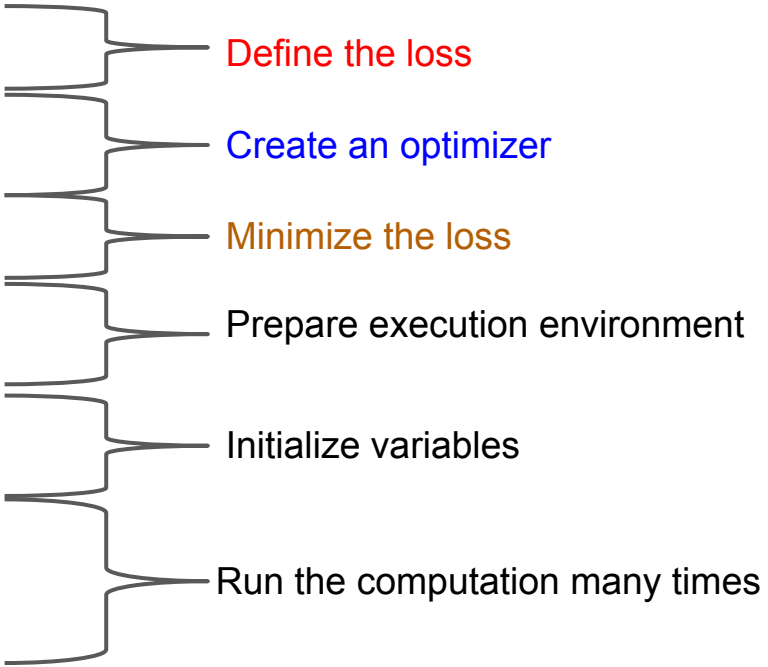
ntop



Information flow

Adding loss, optimizer and training function

```
loss = tf.reduce_mean(tf.square(y - y_label))

optimizer = tf.train.AdamOptimizer(0.2))

train = optimizer.minimize(loss)

with tf.Session() as sess:

        sess.run(tf.initialize_all_variable())

        For i in max_epochs:
                print(sess.run(y, feed_dict={x : x_inputs}))
```



ntop

Results obtained with ConvNet

Used two different datasets, one big (~13K images), one small (200 images)
All the images are cropped or padded in a fix dimension (**224x224x3**)

- **Small Dataset** obtained ***~95%*** of accuracy in prediction
- **Big Dataset** obtained ***- work in progress -***

Thank you for your attention

References:

- **nDPI** github page: https://github.com/ntop/nDPI
- **ConvNet** github page: https://github.com/kYroL01/ConvNet
- **Tensorflow:** https://www.tensorflow.org/

E-mail to **Michele Campus**: campus@ntop.org