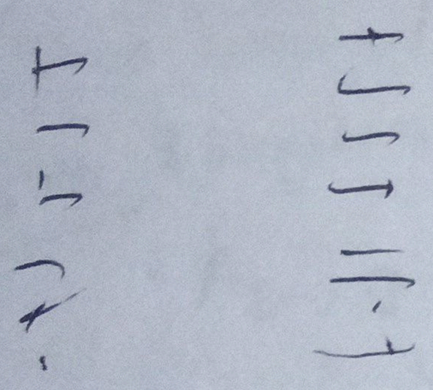
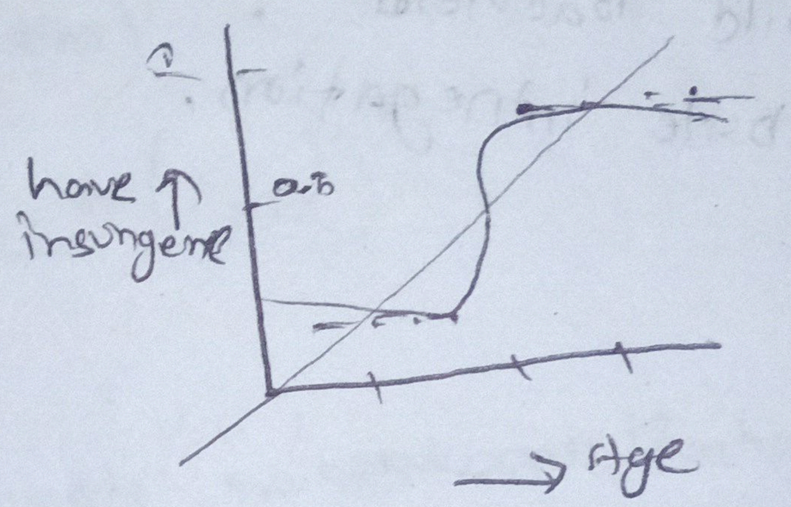
Deep Learning

Why Deep Learning is popular

1. Data growth
2. Hardware advancements (GPU, TPU)
3. Python and Open source Ecosystem (Pytorch, Tensorflow)
4. Cloud & AI Boom

age have_insurance (Binary classification)





sigmoid on logit → S function

$$\text{sigmoid}(z) = \frac{1}{1+e^{-z}}$$

e = Euler's number = 2.718

$\text{sigmoid}(200) \approx 1$ (close)

$\text{sigmoid}(-200) \approx 0$ (close)

$y = mx + c \Rightarrow z = \frac{1}{1+e^{-y}}$

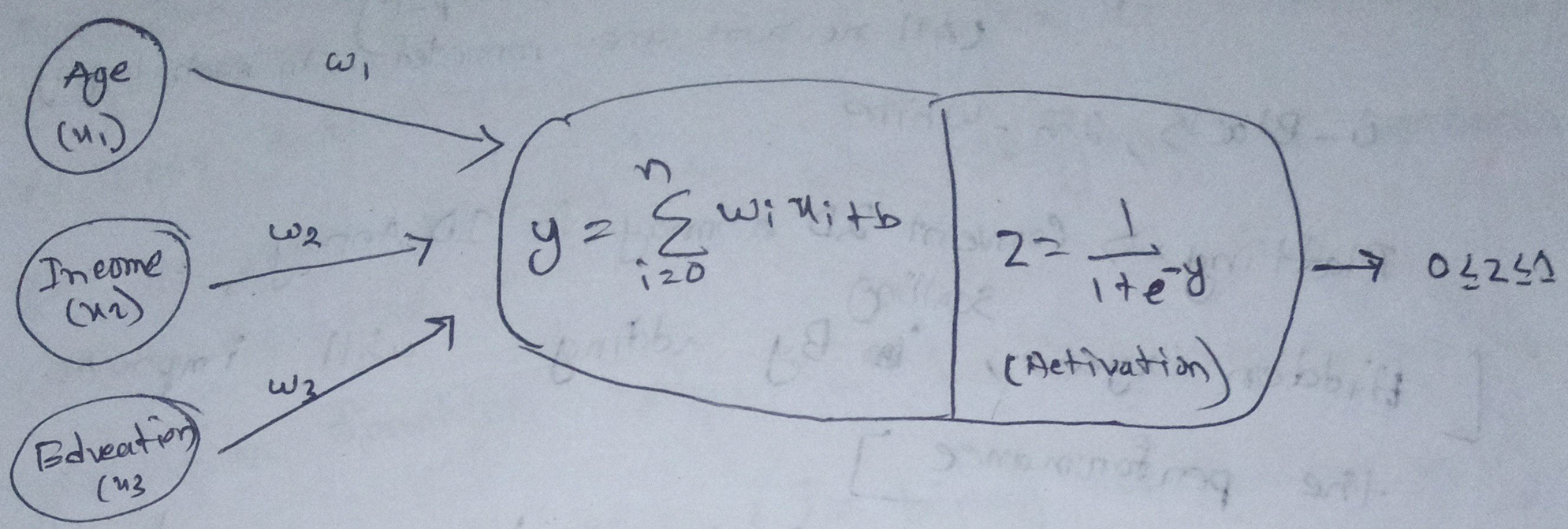
instead of

$$y = \sum_{i=0}^{n} w^i x^i + b$$

logistic regression

⇓

simple like a ~~sig~~ single neuron



A simple neuron

(Forward and Back propagation) => Neural Network simply video

~~Pytorch~~

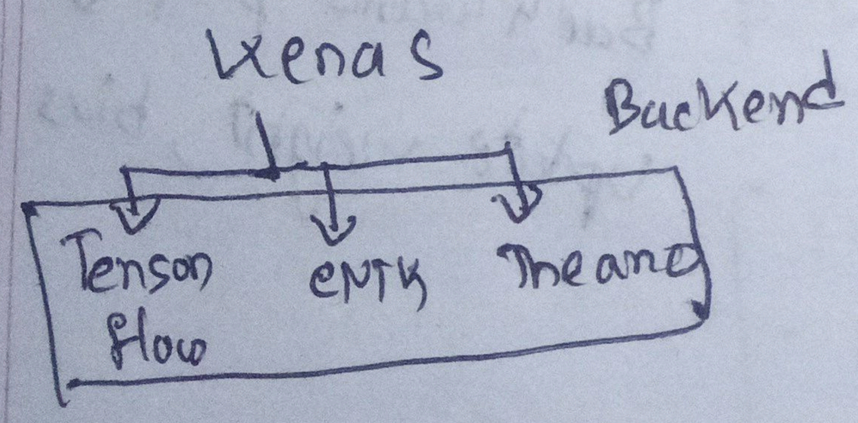
Pytorch VS Tensorflow VS Keras

=> ~~cost~~ Cntk (another framework) , pytorch , tensorflow

microsoft, facebook, Google

Keras → not a full fledged deep learning framework

↳ Nice wrapper



Tensorflow 2.0 → not part of Keras

↓

made part of library itself.

dense layer (Input → hidders → output)
(All neurons are connected with each layer)

0 - Black, 255 - white

Flatting → Convert 2D Array to 1D Array

[Hidden layers, scalling By adding will improve the performance]

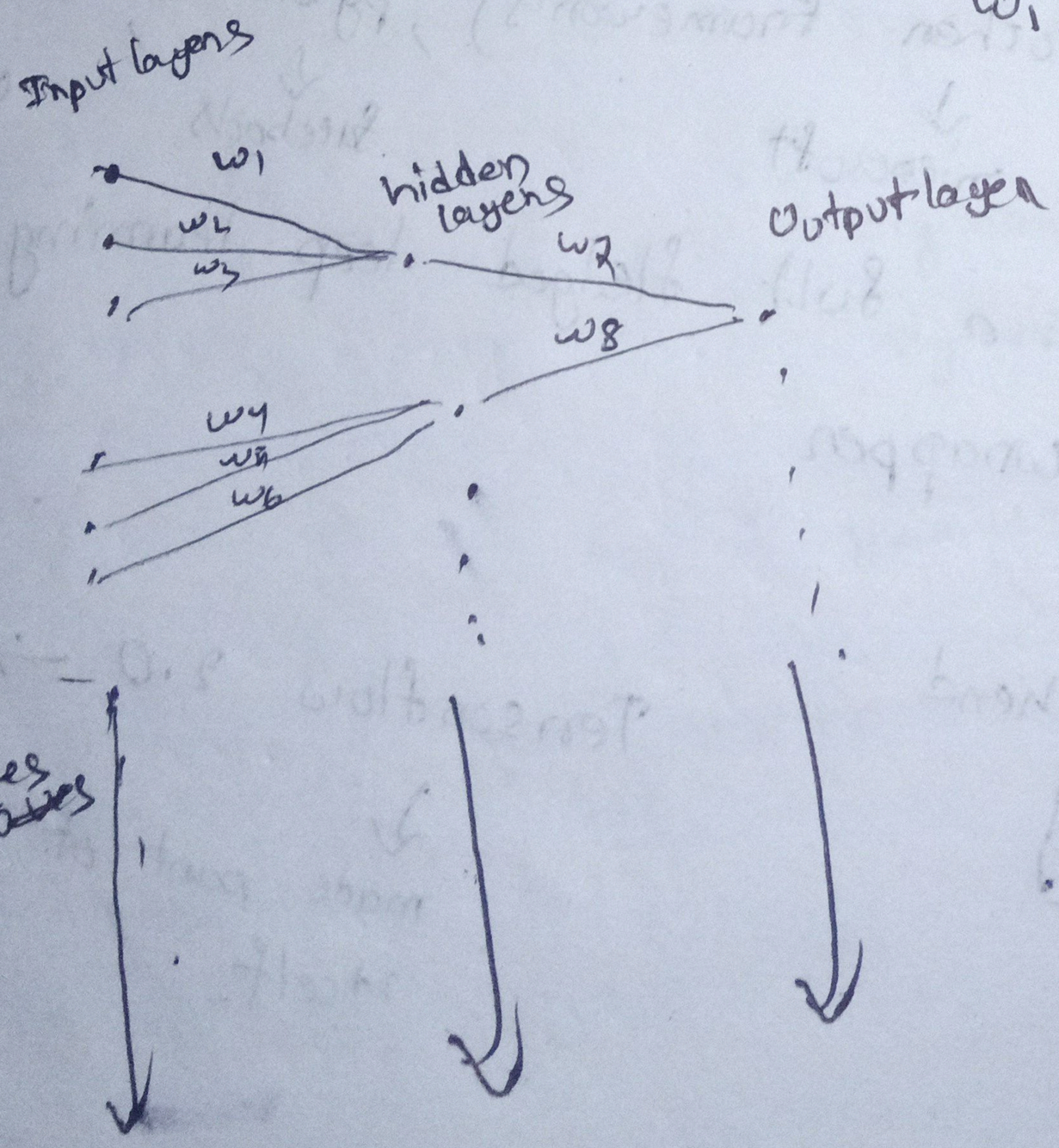
Sequential ⇒ stack of layers in my neural network

optimizers → allows to train efficiently. (To each global optima)

loss function → how different from actual values

sparse → should take integer number,

[Watch confusion matrix]



$w_1, w_2, \ldots w_8$ ⇒ will update through Back propagation.

[Consider bias]

Forward propagation to get result

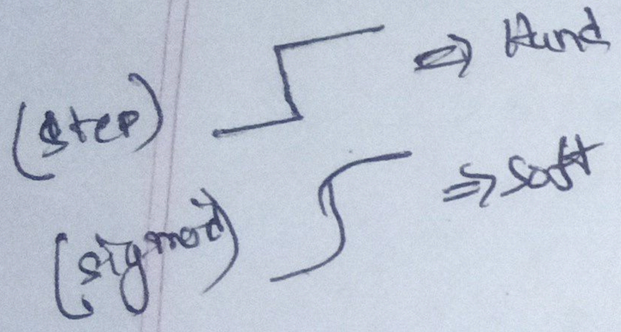
Backward propagation to update weight, bias

Activation function

Without Activation, it will work as linear equation
It is not possible to solve everything in ~~Activation~~ linear equation.

#Activation function helps in building non-linear equation.

Example :
- step Function (For binary classification only)
  - (new version) → sigmoid Function (soft prediction to binary classification)
- tanh Function (To predict -1 on 1)

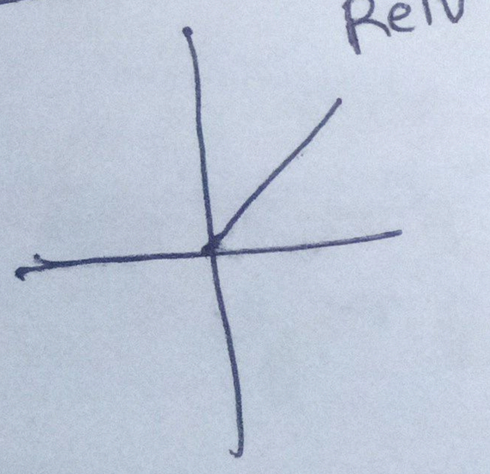


[Use sigmoid in ouput layer. All other places try to use tanh]

Issues with sigmoid and tanh

* derivative = $\Delta y / \Delta x$ [Every points, slope always changes]
* vanishing gradient problem [$\Delta y \cong 0$]

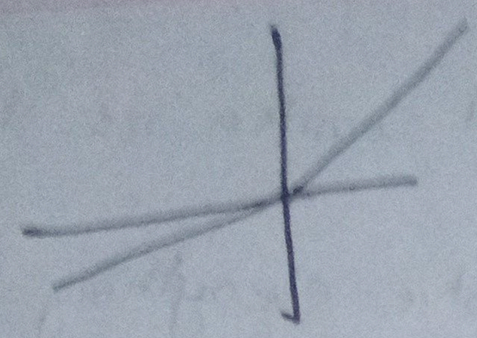
ReLU Function



$ReLU(z) = max(0, x)$

[For hidden layers if you are in confusion what ~~to use~~ Activation function to ~~use~~ use, ReLU ~~will~~ should be applied]

#It also has vanishing gradient problem.



Leaky ReLU

Leaky ReLU = ~~max(0.1x,x)~~
max(0.1x, x)

will not cause not vanishing gradient problem

Slope VS Derivative

For linear equation	For non-linear equation
It is a constant	It is a function

[Learn Derivatives, Matrix, Dot products]

Lost / Cost function (Difference of errors)

(MAE, MSE, Binary cross entropy / log loss

MAE, MSE → a not a function for logistic regression.

Binary cross entropy / log loss → logistic regression function.

Mean Absolute error (MAE)
Mean squared error (MSE)

Gradient Descent ⇒ To find weights (m, b, w, c ... etc)

x	y	
2	7	-0.5
3	9	0
5	13	1
7	17	2
9	21	3

$y = x*2+3$

→ y = ? (Apply Gradient Descent)

$w_1 = w_1 -$ something

learning rate $* \frac{\partial (\text{cost})}{\partial w_1}$

$b = b -$ something

Batch gradient system (Batch = 1)

stochastic gradient systems (Batch = 16, 32, 64 or others)

mini Batch gradient system (Batch = few)

Batch gradient descent

⇒ Run all training samples each one and calculate Σerror, Back propagate to update weights.

⇒ Good ~~small~~ for small training set

Stochastic gradient descent (randomly)

⇒ Run ~~a set of~~ one training samples ~~by group~~ (randomly)

⇒ Good for large training set

Benefits of Both: Mini Batch gradient descent ⇒ Runs a batch of training samples

Chain Rule

Hyperparameters ⇒ controls / affects normal parameters

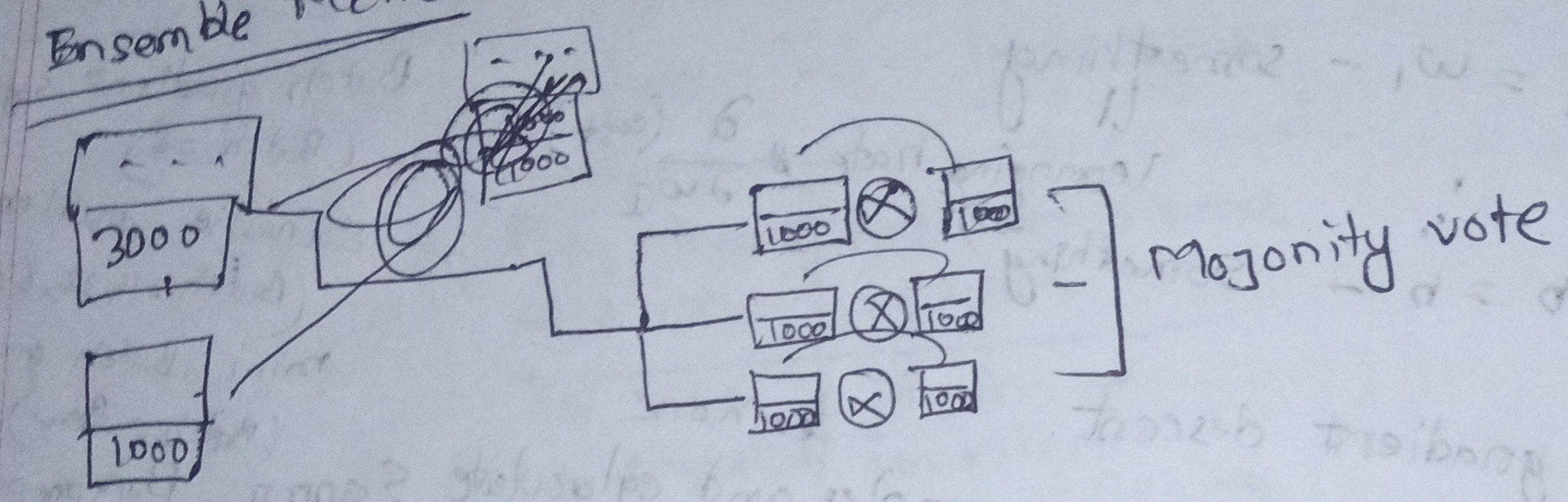
ANN ⇒ Artificial Neural Network

sparse_categorical_crossentropy ⇒ categorical_crossentropy (After One Hot encoding)

Classification metrics #
Dropout / Regularization (Overfitting, Underfitting)
↓
dropout to reduce complexity

Handling ~~imbla~~ imbalanced data ⇒ Under sampling,
Over sampling,
Over sampling (SMOTE),
↓
synthetic Minority Over-sampling Technique

Ensemble Method

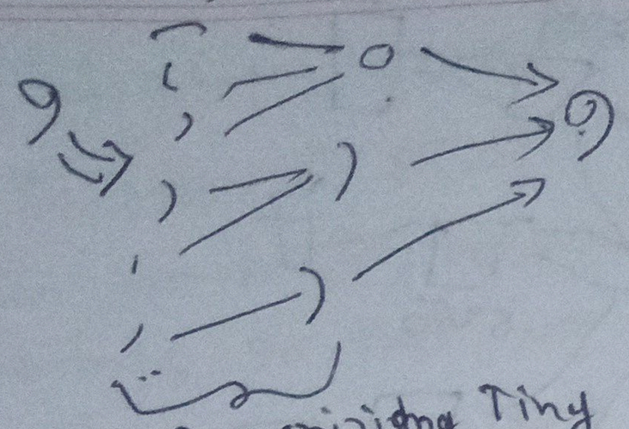


Focal loss
↳ Penalize majority samples and gives more weight to minority class samples, during loss calculation

Application of Computer Vision
#Image classification and object detection
#Banking, Agriculture, Autonomus car, Retail (Amazon go)
CNN (Convolutional Neural Networks)
↳ Recognizing images (we can use ANN but not for bigger image)

Disadvantages ANN for image classification: 1. Too much computation
2. Treat local pixels same as pixels far apart
3. Sensitive to location of an object in an image.

filter detected these features



Recognizing Tiny ~~things~~ features

9 = (loopy pattern, Vertical line detector, Diagonal line detector)

⇒ We Apply convolution operation using filter / kernel

⇓

stride of 1, 2, 3 ...

Filters ⇒ feature detections

Aggregating them together → flattening to vectors [column matrix]

↳ ReLU (Activation functions)

feature extraction

⇓

Then Apply ANN

classification

padding, pooling (Average, Max)

⇓ increase Dimension

⇓ reduce Dimension

Benefits of pooling

⇒ Reduces dimensions, computations,

⇒ Reduces overfitting, parameters (less)

⇒ Makes model tolerant towards variations distortions.

Convolution ⇒ ⊗ (Used to detect features)

- Connections sparsity reduces overfitting
- Conv + Pooling gives location invariant feature detection
- Sharing parameters

ReLU

- Introduces Non-linearity
- Speeds up training, faster to compute

Pooling CNN can not handle Rotation & Thickness (scale)

⇓

You have to do in preprocessing (In data Augmentation also)

⇓

Increasing new samples

~~Introduction to probability~~

~~Christmary's Fixed~~ ~~18.06 Linear Algebra~~

* In convolution, corner pixels ~~as~~ has less ~~important~~ contribution in feature detection

Pooding: so that we do not lose extra information (same input = output)

↳ helps pixels to have more contribution than before

strides ⇒ steps to move in X, Y direction

Transfer learning ⇒ focuses on storing knowledge gaind while solving one problem and applying it to a different but related problem.

saves computational power

Google's trained Model

Almost doing Feature Engineering

[In code, should know feature vector model) → without Top layer

Image classification	V.S	Objects detection	V.S	Image Segmentation
Image classification ~~with~~ on with localization [locating object]		Detecting objects, with the aid of rectangular bounded boxes. (multipling objects)		Classify each of pixels as one of the classes (multipling different objects)
				Instance segmentation semantic segmentation (pixel mask)

3 popular Datasets (ImageNet, ~~fb~~ coco, Google open.)

Sliding ~~windno~~ window object detector

↓ new ~~approach~~ approach

↓ To get bounding ~~to~~ boxes what will be the sizes?

↓ Try different sizes (small to langes)

R CNN

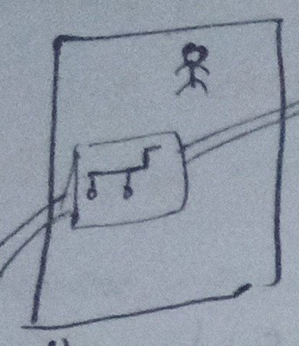
1) (certain area based on Image segmentation)

↓ Fast CNN (Fast speed)

↓ Faster CNN (Faster ~~CNN~~ speed)

↓ YOLO (You only look once) with different versions

YOLO (You only looks once) [It is very fast]



= Output
Dog = 0
person = 0
+
Bounding Box

⇒ It works only for single objects, what about multiple objects

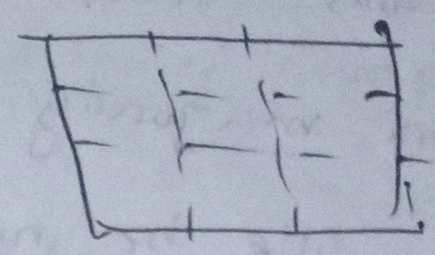
⇓

Divide the image into grid cells

Vectorized form

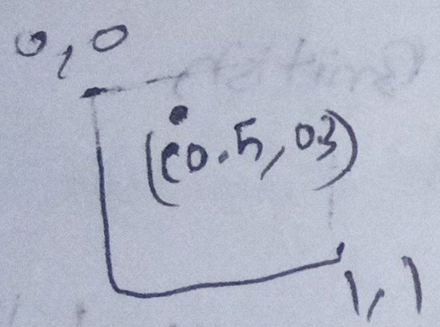
⇓ vector

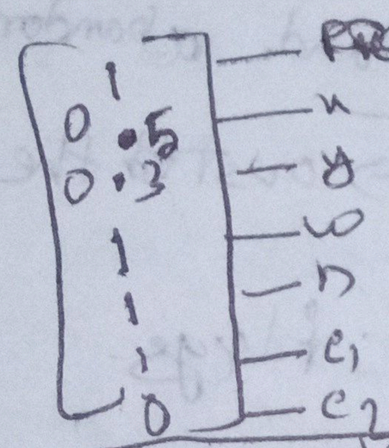
Pc	1
Bx	50
By	70
Bw	60
Bh	70
c_1	1
c_2	0



Grids	Vectors
	[]
	[]
	[]

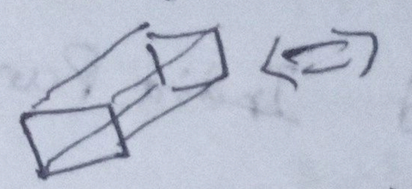
c_1 = Dog class
c_2 = person class
Pc = Probability of of c_1 or c_2
Bx, By = Bounding box x, y position
Bw, Bh = width, height of Bounding box

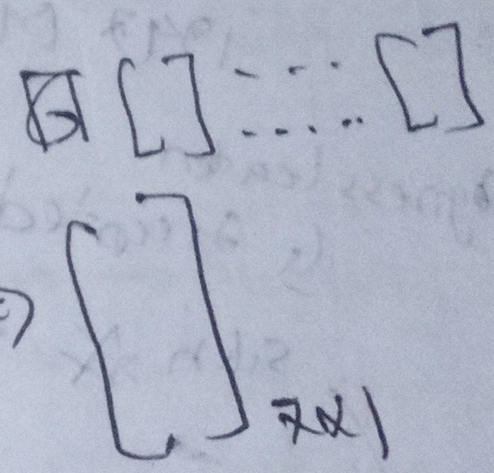




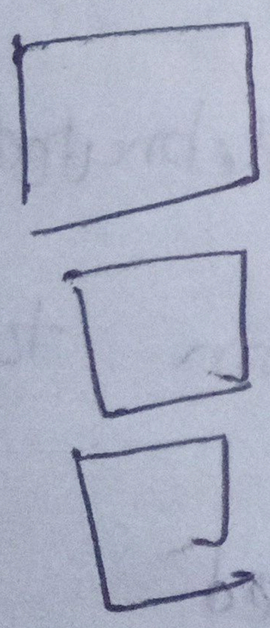
Target vector
y_train (m×n such vectors)

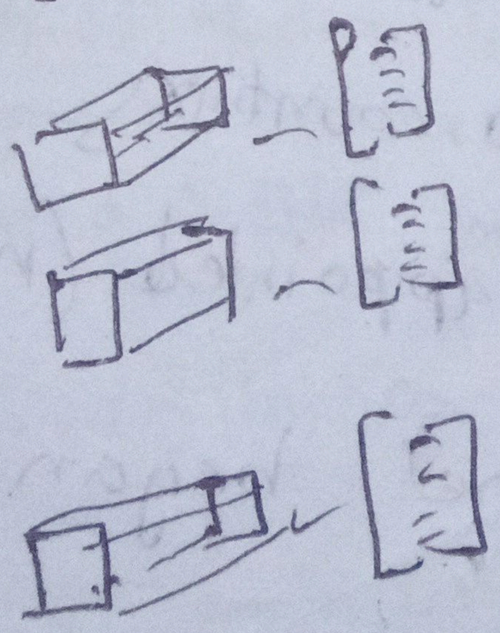
4 by 4 by 7





X_train





Issues

Multiple Bounding rectangles (can take the max? ⇒ will lose information)

can use IOU ⇒ Intersection over union (Intersect / union)

Try to find overlapping area.
1 ⇒ highest overlapping
0 ⇒ no overlapping

pick threshold IOU = 0.6 or 0.5

August 9th → ~~Radcliffe~~ Radcliffe finish ... then

Lorone was mixed now it is only muslim.

Issues of YOLO

Multiple Bounding Boxes

["can we take maxes? if we take we might lose information]

we can use IOU (Instersection over union)

IOU = intersect area / union area

⇓ Try to find overlapping area (set thershold 0.6 or 0.7)

Discard those below the thershold values.

Non max suppnesion (Get the max probability between over lapped areas)

What if one grid cells has center of two objects.

⇓

concat these vectors of n objects

⇓ Anchor boxes