**CS231n deep larinng and computer vison**

**Lecture 1:**

-530 million years ago sight was first developed.

- in the 16th century Leonardo da vinci the first camera (camera Obscura).

-story of the cat and visual processing.

- blocks world Larry Roberts 1963 was trying to extract engels of an objects.

- john McCarthy is responsible for the term AI.

-summer of 1966 the birth of computer vision in MIT.

-david mark: each layer is more complicated than the previous one.

-the number of pixels in a 10mp img > the total number of items.

**-perceptual grouping:** carv specific objects from an image.

**Lecture 2:#H=hyperparameters**

-problems like **image classification** is brittle to solve by just hard coding it .

-the **Challenges;**

* **Viewpoint variation, Scale , Deformation, Background clutter**
* **Occlusion;** only a small portion of an objec could be visible.

-trainng time can be high but testing should be low like o(1)...

-**nearest neighbor classifier**: can be used to image classification but is bad When uesing is **k(H)** shouled be bigger than 1.

-The choice of distance(H): what makes two imgs(arrays) neighbors.

-not used in imgs

**Settting hyperparameters:**

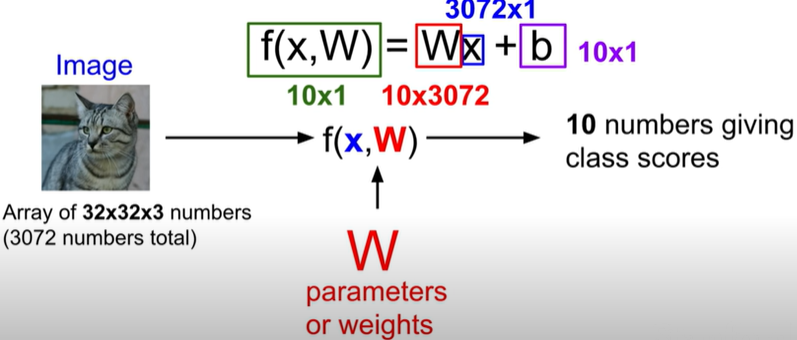
**method 1:**

-to use the best hyperparameters you need to keep evaluating your modul until you get the best evaluation

We use [train ,validation,test] data only use the test set to give the final accersy of the module.( validation set : test to choose hyperparameters).

**method 2:** cross-validation

**Linear Classification:**



-we have 10 class (to classify images) to use in the function f() the higher result means that it belongs to this class

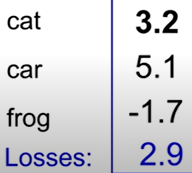
-W=10 rows each one represent a class ,The problem we have only one template for each class

**Lecture 3:**

-**Loss function(Li):**

**1-Multiclass Svm**: loss in one test image.

the property of a cat is 3.2 loss=max(**0,** 5.1-3.2+1)+



max(**0,** -1.7-3.2+1)=2.9+0=2.9 (in max choose between 0 and the sum)(the +1 is bios)

modul loss =is the average lost in all test images.

- loss function is the way to tell your modul what is the type of error I care about.

**Li=data loss+Regularizarion**

**-Data loss:**predictions must match training data(using only this our modul while overfit)

**-Regularizarion:**used to Reduce Overfitting :we let go of(fitinng training data) to get more (generaliza on unseen data). (λ(H): Regularizarions strength ).

-used to penalizing for complexity,**Typs:**

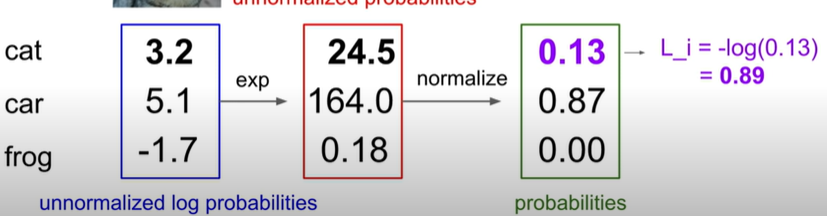
* **ML and DL** : L1 ,L2 ,Elastic net(L1+L2) ,max norm.
* **DL** : Dropout , batch , stochastic depth

- Occam’s Razor : Among competing hypotheses the simplest is the best.

-how do we choose the best reg?

**2-Softmax classifie**: by using softmax function we well have probabilities of each class.

**Loss function :** Li=-log p(Y=yi|X=xi) the probabilities of yi is high and close to 1.



**How do we find the best way to minimize the loss?**

**Optimization:** by using gradient descent(I know it so no need to explain it agin here).

-each optimizer uses a different gradient descent (in general it is still the same).

**Lecture 4:**

**Computational graphs(CG):**a graph to compute Li.

**Backpropagation:** for every training image we modify the wighets a bit , doing this to all training images and averaging the (modification) is called a gradient descent **step(adjust the weights)**. Source : <https://www.youtube.com/watch?v=Ilg3gGewQ5U&ab_channel=3Blue1Brown>

-if we use only linear layers(in our N-N)the result is not different from using one large linear layer, because of this we you nonlinear functions/layers

-the idea of N-N is stacking(in a hierarchical way) different functions on top of each other so we have a more complex nonlinear function.

**Why should we use nonlinear functions/layers:**

-as we said before in the first layer we only have **w1**=x number of unconnected images but in the nonlinear layer **w2**= we connect multiple images under the same class , we can say these two rows of w (two images) are in fact both represent a car yellow one and a blue one .

For example: in w2 we can decide which images in w1 should have the highest effect on the final value of thes pretcler class.

Activation functions:[sigmoid,tanh,ReLU,Leaky ReLU,Maxout,ELU] just a preview.

**Lecture 5:**

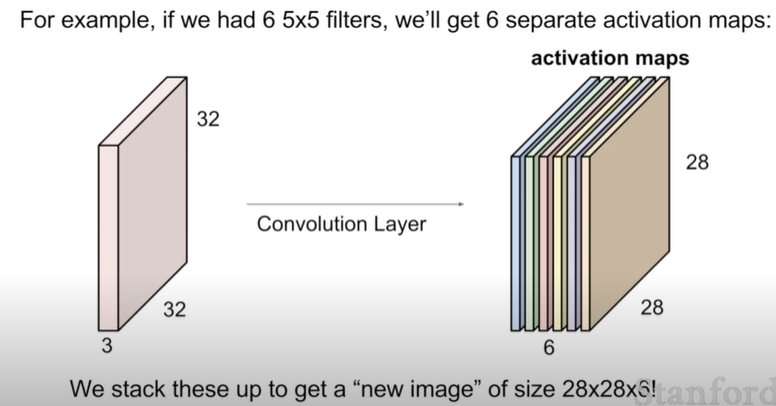
**Convolutional Neural Networks(ConvNets):** in n-n we turn the input image (32,32,3) into a (3072,1) vector but in CNN the input stays In the shape of (32,32,3).

-Instead of weights we will have filters(5,5,3):**3 channels is the same as the input**

**-activation map/** **feature map:** the product of the filter and a piece of the image= piece of the activation map.

-keep sliding and multiplying until we have a sheet of results(28,28,1)

-stack multiple activation Maps of different filtersthe more CNN layers the more complex activation maps/features.



**Pooling layer (max poling):** downsizing the width and height but depth stays the same

Fc-layer=dense layer

**Lecture 6:**

**Training Neural Networks**

Saturated Sigmoid Neurons: A logistic neuron is said to be saturated when it reaches its peak value either maximum or minimum.

**Activations :**

* **Sigmoid:** Force the values to be between 1 and 0 (firing rate or probability)

**Disadvantages: 1-**Saturated Sigmoid Neurons will kill the gradient(no more steps)**2-not zero centerd**

* **Tanh():**Force the values to be between -1 and 1.(zero centered) but still kills the gradient as before
* **ReLU:**if negative x=0 elif positive x=x.( no Saturat if positive ,computationally efficient, converge faster than the above functions,not zero centered) dying radio problem: the gradient is killed and accersy stagnated
* **Leaky ReLU**:same as the Relu but we initialize the bias =0.01 so it don’t stagnat(well not die)
* **Elu:** all benefits from RuLu
* **Maxout”neuron”**

Use Relu try out the othors but don’t use sigmoid.

**Data preprocessing:** in images we only use zero-centered data