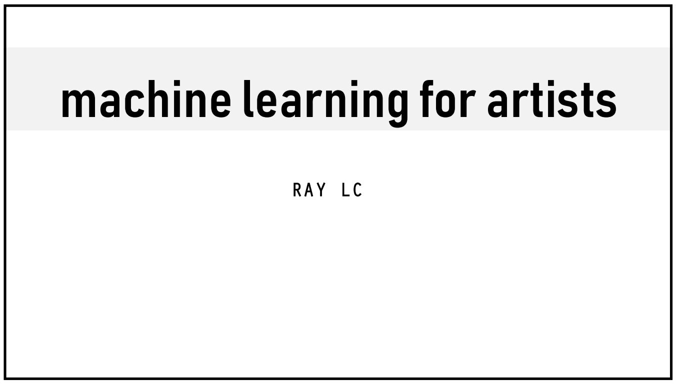
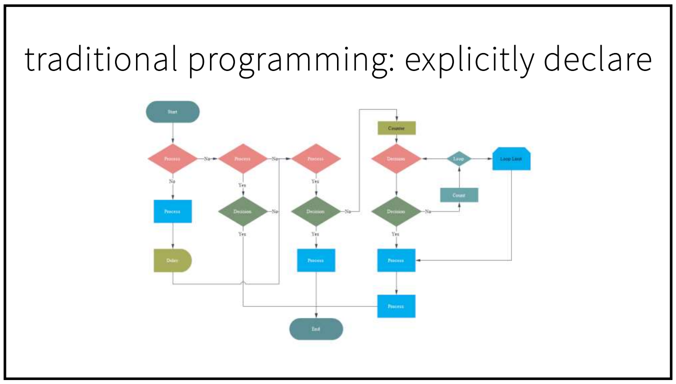
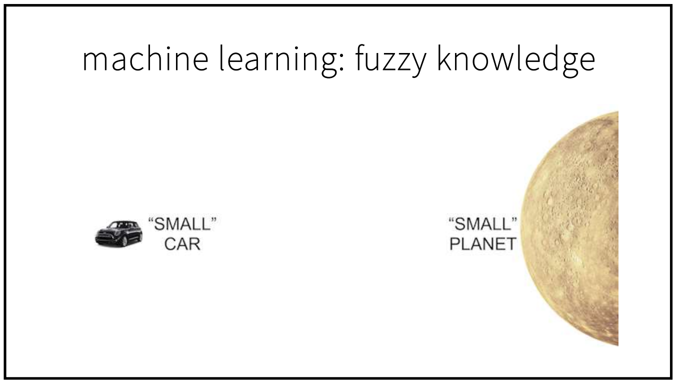


machine learning for artists

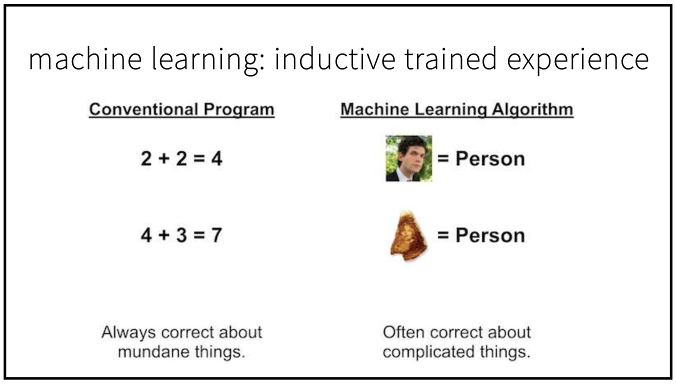
RAY LC



traditional programming: explicitly declare

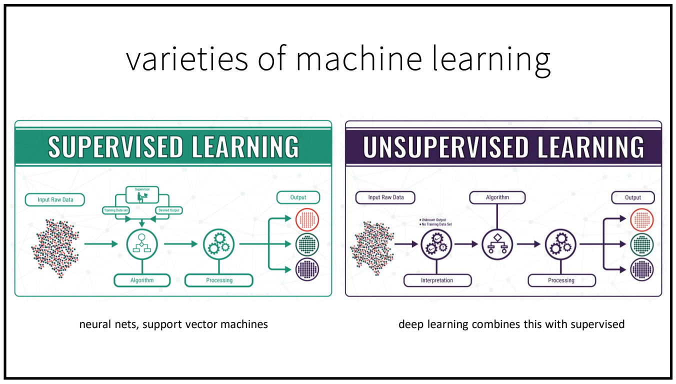


machine learning: fuzzy knowledge



machine learning: inductive trained experience

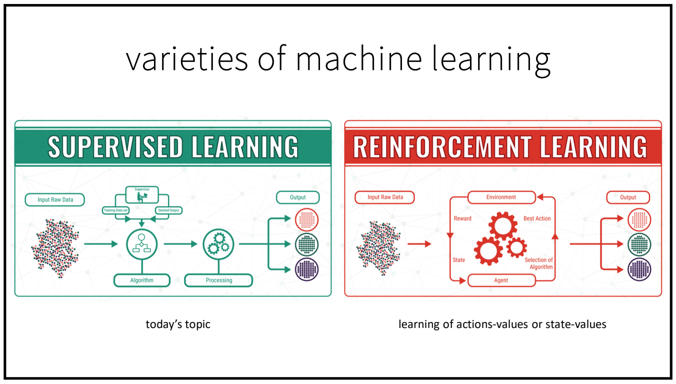
Conventional Program	Machine Learning Algorithm
2 + 2 = 4	= Person
4 + 3 = 7	= Person
Always correct about mundane things.	Often correct about complicated things.



varieties of machine learning

neural nets, support vector machines

deep learning combines this with supervised

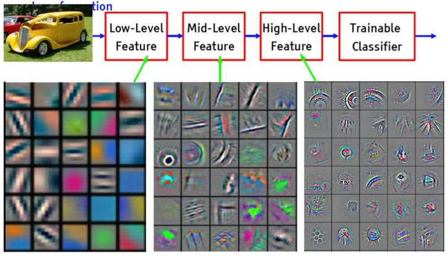


varieties of machine learning

today's topic

learning of actions-values or state-values

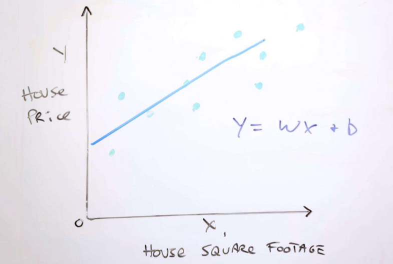
Deep Learning = Learning Hierarchical Representations



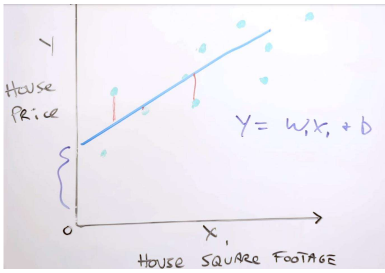
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

9

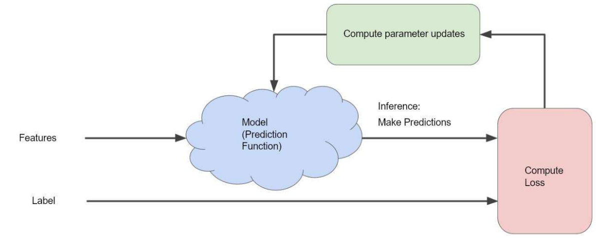
so what is supervised machine learning?



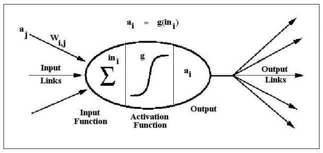
move the line to minimize errors

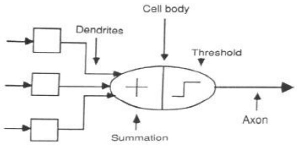


supervised machine learning workflow

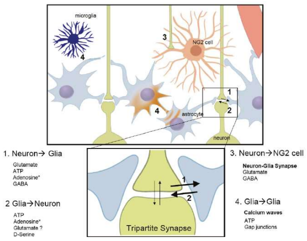

Compute parameter updates
Features
Model (Prediction Function)
Inference: Make Predictions
Label
Compute Loss

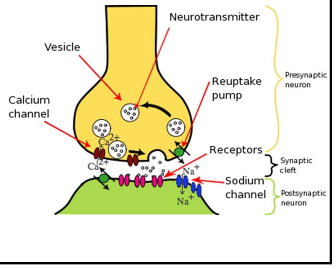
single layer perceptron vs human neuron



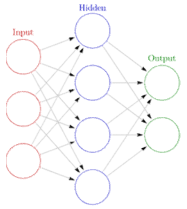


in reality in the human neuron





multilayer neural network



$$\Delta w = w - w_{old}$$

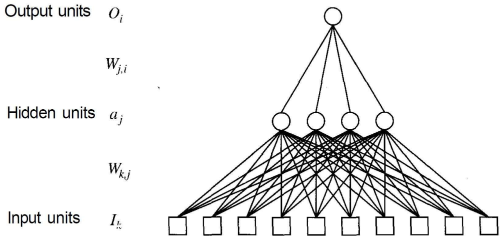
$$= -LearningConstant\frac{\partial E}{\partial w}$$

$$= (LearningConstant)(y_{target} - y)(x)$$

or…

$$w = w_{old} + (LearningConstant)(y_{target} - y)(x)$$

multilayer neural network: credit assignment


Output units O_i

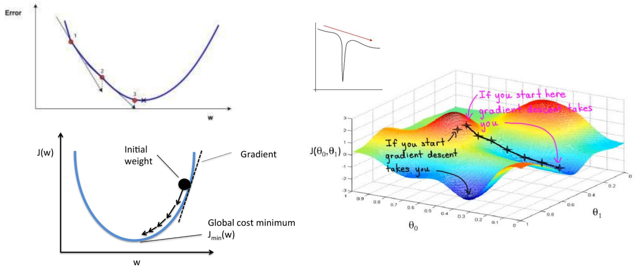
$W_{j,i}$

Hidden units a_j

$W_{k,j}$

Input units I_k

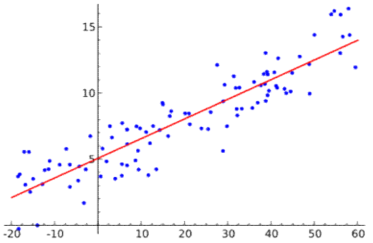
gradient descent in weight space



weight updating using backpropagation

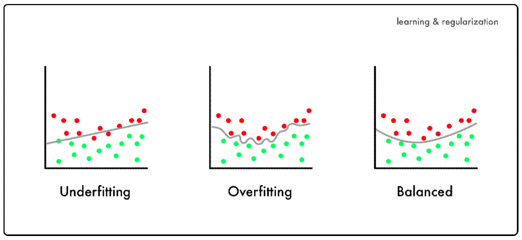
```
repeat
  for each e in examples do
    /* Compute the outputfor this example *I
      O ← RUN-NETWORK(network, $\mathbf{I}^e$)
    /* Compute the error and Δ for units in the output layer */
      $\mathbf{Err}^e \leftarrow \mathbf{T}^e - \mathbf{O}$
    /* Update the weights leading to the output layer */
      $W_{j,i} \leftarrow W_{j,i}$ + o x $a_j$ x $Err_i^e$ x $g'(in_i)$
    for each subsequent layer in network do
      /* Compute the error at each node */
        $\Delta_j \leftarrow g'(in_j) \sum_i W_{j,i} \Delta_i$
      /* Update the weights leading into the layer *I
        $W_{k,j} \leftarrow W_{k,j}$ + a x $I_k$ x Ay
    end
  end
until network has converged
```

multilayer net for statistical regression



another encoding needed for categorical data

over generalization



mitigate using unsupervised learning (nonspec patterns)

and now, your exercise

- find photos online that you want to use ML to recognize.
- use ml5.min.js package and an online engine to classify them.
- let's write classifyimg.html together (or just use mine).
- run local SimpleHTTPServer.
- in js (given you), the call is:

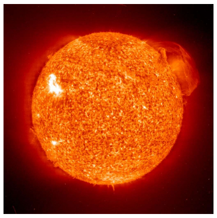
```
classifier = ml5.imageClassifier('MobileNet', function()
    console.log('Model Loaded!');
});
```

then call classifier.predict to make classifications.

CONTEST: look for images that machines most likely will get wrong.

example images found on web

dishwasher: easier for machine than humans



sun that got classified as orange: harder for machines

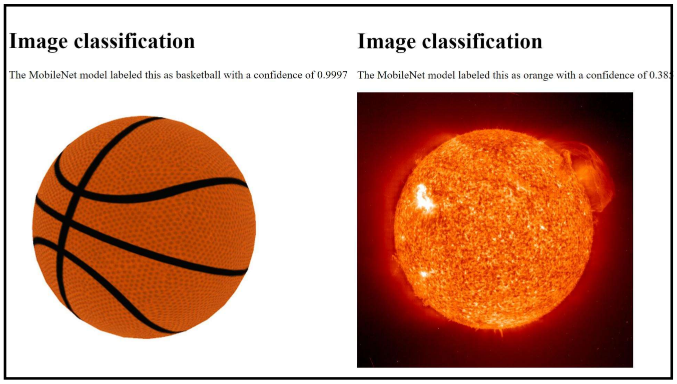


Image classification

The MobileNet model labeled this as basketball with a confidence of 0.9997

Image classification

The MobileNet model labeled this as orange with a confidence of 0.38

machine learning for artists

RAY LC