CS 331
Computer Vision

09 / 30 / 2024

Instructor: Michael Eckmann

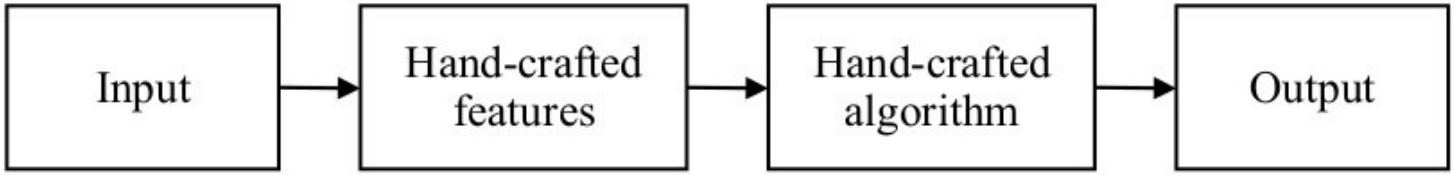
Today's Topics

- Questions? / Comments?
- Nearest Neighbors

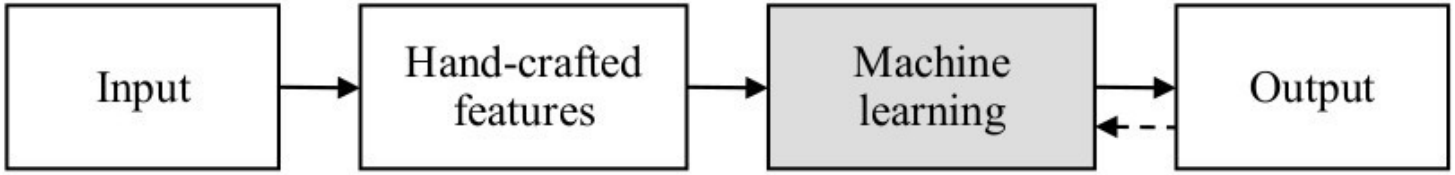
Switching gears

- Start Chapter 5 Deep Learning in Szeliski
  - reading: pages 236 – middle of 243
- Supervised learning
- Nearest Neighbors

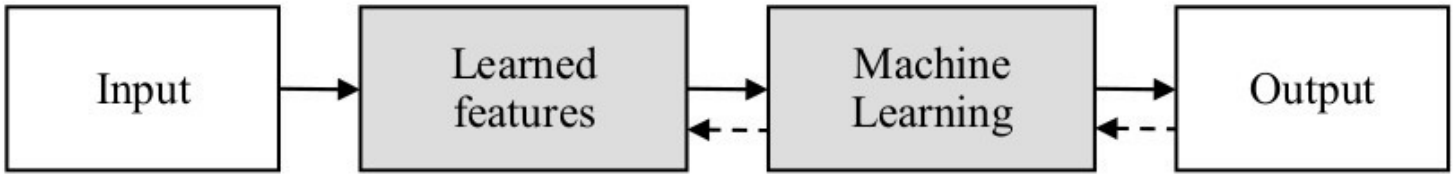
CV pipelines



(a) Traditional vision pipeline



(b) Classic machine learning pipeline



(c) Deep learning pipeline

Supervised Learning

- Specifically for classification tasks.
- Supervised learning is where both the inputs and the known labels of the inputs are given
- e.g. images of cats are labelled with the label cat and pictures of dogs are labelled with the label dog and the images and their known labels (cat or dog) are provided to the learning algorithm
- Differentiate Supervised learning from Unsupervised learning
- Classification definition – the working system will classify an unseen input to a hopefully correct label
- e.g. an image of a cat that wasn't seen during training will hopefully classify the image as a cat

Supervised Learning

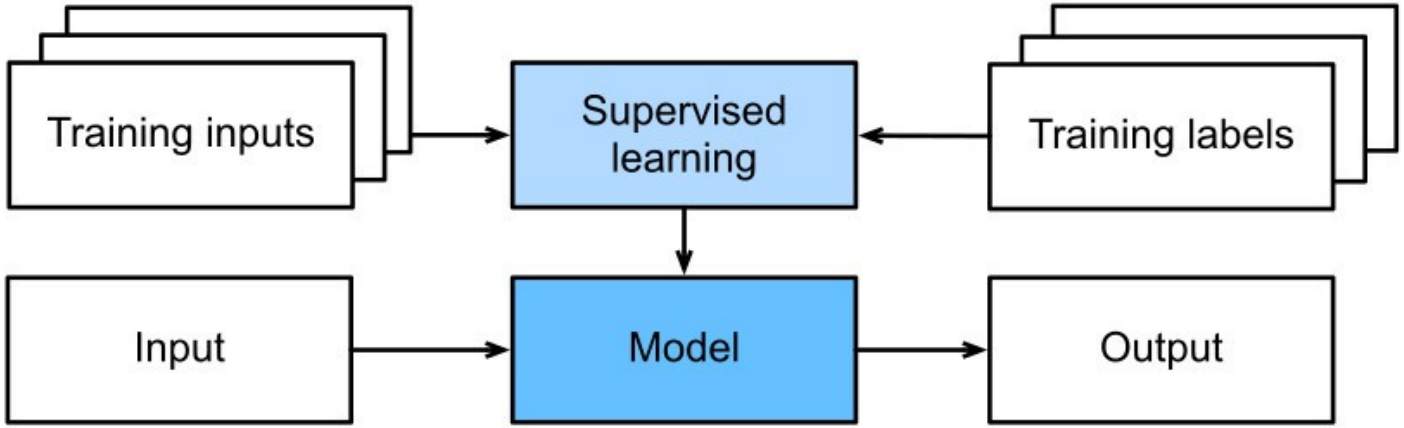


Figure 5.3 *In supervised learning, paired training inputs and labels are used to estimate the model parameters that best predict the labels from their corresponding inputs. At run time, the model parameters are (usually) frozen, and the model is applied to new inputs to generate the desired outputs. © (Zhang, Lipton et al. 2020, Figure 1.3)*

Supervised Learning

- Training a ***supervised learning*** system
- Provide all the ***training*** data
- Includes *input samples* and their known *target labels*
- Output a ***trained model***
- Training includes ***minimizing some loss function*** (a function whose value is large when input samples are misclassified (to a wrong target label) and small when most input samples are classified correctly.
- ***Testing***
- Use the trained model to ***predict*** outputs (target labels) for new inputs (unseen during training)

Data Preprocessing

- Prior to training and testing, data preprocessing usually takes place.
- Some typical preprocessing that is done:
  - Crop or scale all images to same number of rows/columns
  - Centering - computing the mean of all data and subtracting the mean from each input so that the mean of the new set will have 0 mean.
  - Standardizing - rescaling so that variance of the data is 1.
  - Whitening - make it so that the various dimensions of your data are uncorrelated to each other

Hyper-parameters and learnable parameters

- ***Hyper-parameters*** are those variables that we set to fixed values before we train and test our model
- ***Parameters*** (aka ***weights*** or ***learnable parameters***) are those variables that are learned during training

Nearest Neighbors

- Non-parametric --- no ***learnable parameters*** for the model during "training"
- K-Nearest Neighbors – Value for K specified to a fixed value ahead of time and the Nearest Neighbor classifier will, given a new input, determine which k training samples are nearest to the input.
- K is the only ***hyper-parameter*** for k-nearest-neighbors.
- To determine nearest, use some distance measure.
- The k training samples that are nearest are then processed in some way to produce an output (e.g. they can be averaged or all k of them output as answers)

Nearest Neighbors

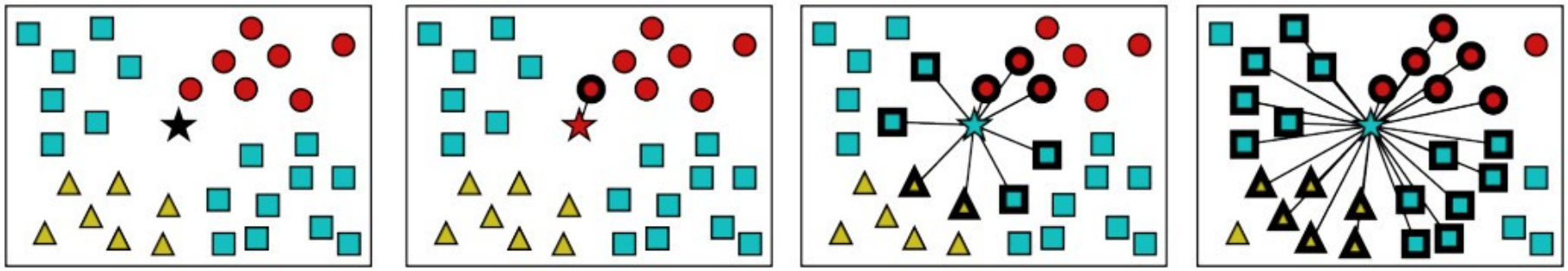


Figure 5.4 *Nearest neighbor classification. To determine the class of the star (★) test sample, we find the k nearest neighbors and select the most popular class. This figure shows the results for $k = 1$, 9, and 25 samples.* © *Glassner (2018)*

Nearest Neighbors

- The distance between input data and every "training" sample can be computed and the samples with the k lowest distances are output.
- The determining of the k outputs given some input is the testing phase here, and can be compute intensive if you have tons of samples (because distance needs to be computed between the input and each of the samples).
- Approximate nearest neighbors techniques are known and will be much less compute intensive but are approximate.
- K-NN is an algorithm that is unusual in that nothing is done during training (takes 0 time), and testing is expensive (takes a long time). For other algorithms, it is usually the other way around.

Nearest Neighbors

- The text mentions (on page 242) a few very important concepts like *decision boundaries, underfitting, overfitting,* and *cross-validation* that I feel are best defined when we get to more complex algorithms where we can really see how they play out