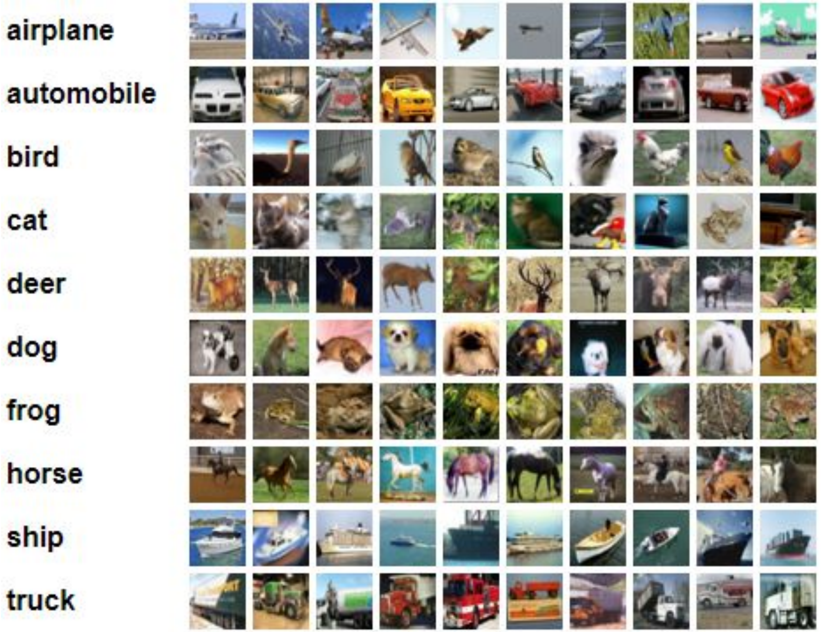
Comparison of Supervised Learning Methods

Machine Learning F21

Andreas Daasbjerg, Andreas Poulsen
VGIS7
adaasb18@student.aau.dk, ampo18@student.aau.dk
16/11-21

Background

- Compare the supervised learning methods from the course
- By method
- By experiments
- Cifar-10 dataset
  - 50000 training samples
  - 10000 test samples
  - 10 classes
  - 3072 features



https://www.cs.toronto.edu/~kriz/cifar.html

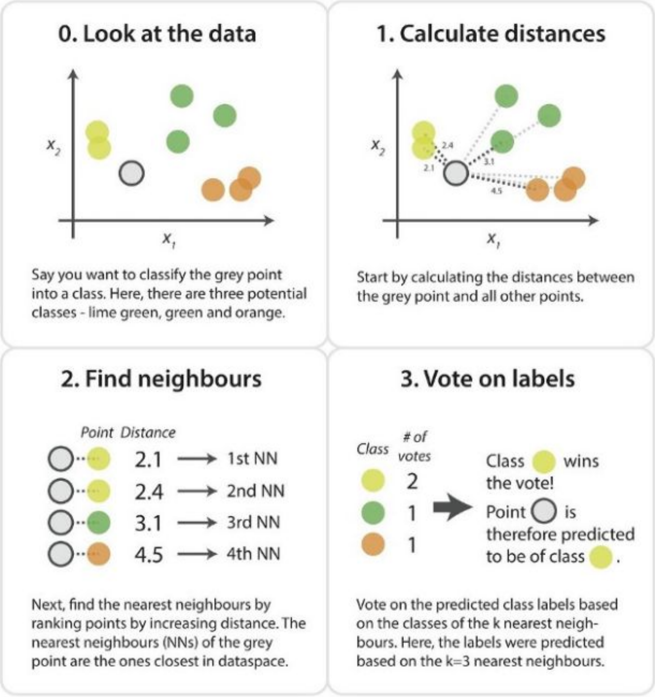
Background

- K-nearest Neighbors
- Decision Tree
- Linear Discriminant Analysis
- Support Vector Machines
- Neural Networks

Methods

- Dimensionality reduction through PCA
    - Proportion of Variance > 90%.
    - 3072 dimensions → 99 dimensions.

- K-Nearest Neighbors
    - Distance measure to classify sample.
        - Euclidean, Manhattan, Mahalanobis Distance.
    - Finding the optimal K-value.
    - Lazy Classification method.



0. Look at the data

Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

1. Calculate distances

Start by calculating the distances between the grey point and all other points.

2. Find neighbours

2.1 → 1st NN

2.4 → 2nd NN

3.1 → 3rd NN

4.5 → 4th NN

Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

3. Vote on labels

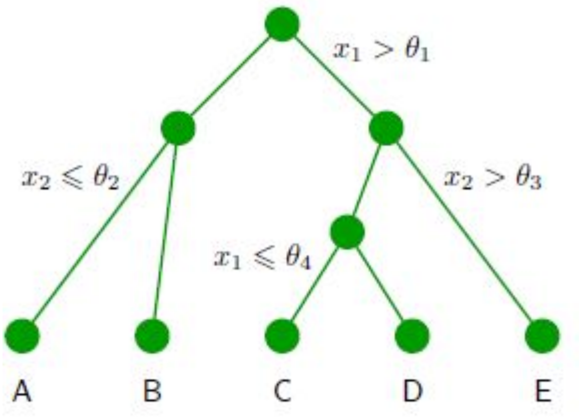
Class wins the vote! Point is therefore predicted to be of class .

Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.

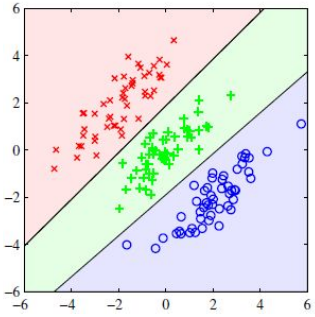
https://www.kdnuggets.com/2016/01/implementing-your-own-knn-using-python.html

Methods

- Decision Trees
  - Divides input space into two regions, ≤ **θ** or > **θ**.
    - Split criterion
    - **θ**-parameter of the model (threshold parameter).
    - Assign each region to a specific class.
- Linear Discriminant Analysis
  - Assigning an input vector **x** to one of K discrete classes
  - Input space divided into decision regions with decision boundaries



Chris Bishop, Pattern Recognition and Machine Learning, Springer, UK, 2006

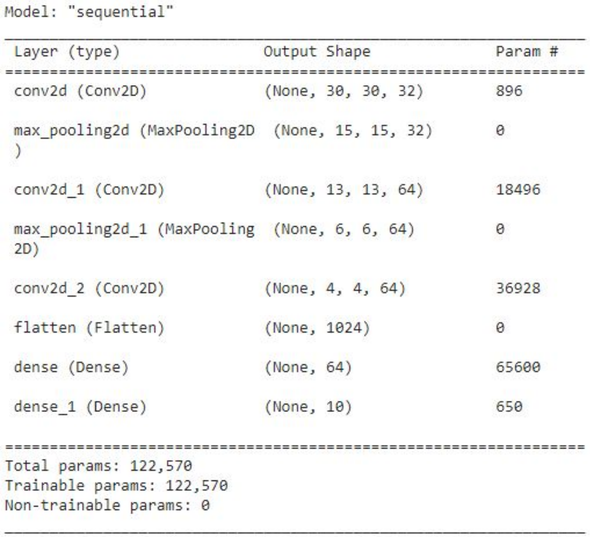


Chris Bishop, Pattern Recognition and Machine Learning, Springer, UK, 2006

Methods

- Linear Support Vector Machine (LSVM)
  - The determination of the model parameters corresponds to a convex optimization problem.
  - Decision machine
  - Maximal margin, Dual Space, Kernel Trick
  - For the linearly separable case, the support vector algorithm simply looks for the separating hyperplane with largest margin.

- Convolutional Neural Network
  - Assumes that Inputs are image
    - Conv2d: ReLU filter.
    - Max_pooling2d: Downsampling 2x2 pool size
    - Flatten: Flatten matrix to 1-D
    - Dense: Reduce tensors to amount of classes in dataset.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 30, 30, 32)        896

 max_pooling2d (MaxPooling2D  (None, 15, 15, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 13, 13, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 conv2d_2 (Conv2D)           (None, 4, 4, 64)          36928

 flatten (Flatten)           (None, 1024)              0

 dense (Dense)               (None, 64)                65600

 dense_1 (Dense)             (None, 10)                650

=================================================================
Total params: 122,570
Trainable params: 122,570
Non-trainable params: 0
_________________________________________________________________
```

Experiments

- K-Nearest Neighbor:
    - Computation of around 4 hours to find best k-value:
        - Fitting data from 2 clusters to 50000 clusters
        - Visualising error rate vs. K-plot would have given the optimum k-value.
            - Choosing the k-value giving the least amount of error.
        - Unknown error → Value never calculated.
    - K = 5, gave 38.15% accuracy.

- Decision Tree:
    - Opted to find the depth giving the best accuracy.
    - Accuracy ~31.5% with max depth of 8.

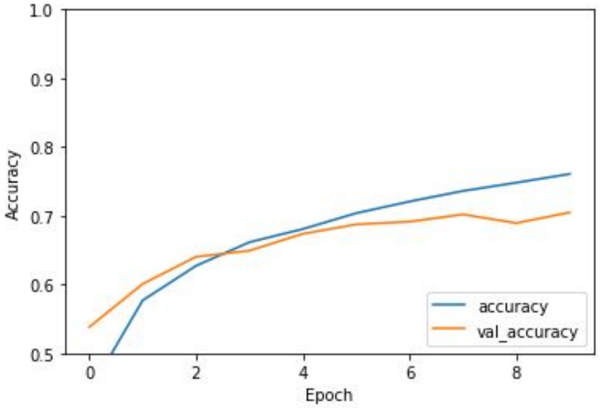
Experiments

- Linear Discriminant Analysis
  - Fitting the model on dimensionality reduced data (3072 → 99)
  - Yielding accuracy of ~40%

- Linear Support Vector Machine (LSVM)
  - Initializing SVM with linear kernel
  - Fitting on data with reduced dimensionality (99)
  - Took too long to fit, so no results yielded

Experiments

- Convolutional Neural Network (CNN)
  - Made with sequential model in TensorFlow
  - Trained for 10 epochs
  - Best accuracy 70.48%



Discussion - Methods

- Since the features of the dataset is only pixel-values, the classes could be overlapping
- Thus, linear classification models could have a hard time discriminating them

- KNN calculates distance to closest samples and classifies based on majority - mixed classes result in bad accuracy

- CNN is great for images because of the convolution layer
- Detects patterns in images

Discussion - Experiments

- CNN accuracy at 70%, substantially better than 40% from LDA

- All methods were tested in Google Colab with a maximum continuous runtime of 12 hours

- SVM not being able to compute could be the result of its quadratic programming, or it not dealing well with large sample sizes

- Estimating K, proved to take too long when computing from 2 → sample_size.
- KNN underperforms because of few features.

Conclusion

- CNN is good for image classification as a supervised learning method
- Linear models are not the greatest choice, if no features are extracted from the images beforehand