MACHINE LEARNING WORKSHOP

Introduction to Classification

2025-27-04

TECHNISCHE UNIVERSITÄT DARMSTADT

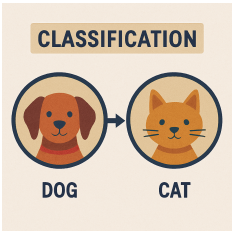
CONTENTS

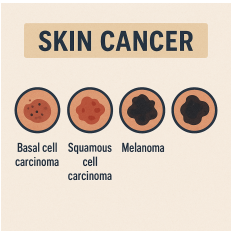
INTRODUCTION

Section 1

WHAT IS CLASSIFICATION?

- Categorizing data into predefined groups
- Examples:
  - **Spam detection**: Email -> Spam or Not Spam
  - **Image recognition**: Picture -> Cat or Dog
  - **Medical diagnosis**: Symptoms -> Disease Type





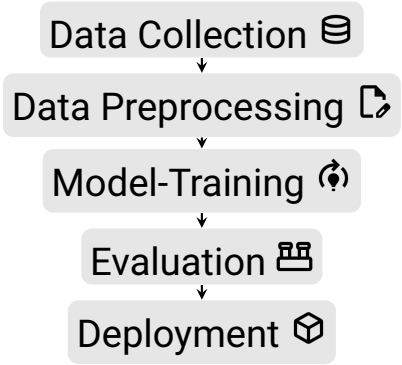
INTERACTIVE: BE THE CLASSIFIER!

Task: Look at these emojis and classify them as ”happy” or ”sad”.

Discussion: What **features** did you use?

TECHNISCHE UNIVERSITÄT DARMSTADT

MACHINE LEARNING WORKFLOW



CIFAR-10 DATASET

Section 2

INTRODUCTION TO CIFAR-10

What is CIFAR-10?

- A dataset of **60,000 images** in **10 classes** (airplanes, automobiles, birds, etc.).
- Each image is **32x32 pixels** and is color-coded (RGB).
- **Classes**: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck.

CIFAR-10 IMAGES AND LABELS

Goal: Classify these images into one of the 10 categories.

STOP 1: EXPLORE CIFAR-10

Click on run in the **part1_explore_cifar10.py**

CLASSIC CLASSIFICATION

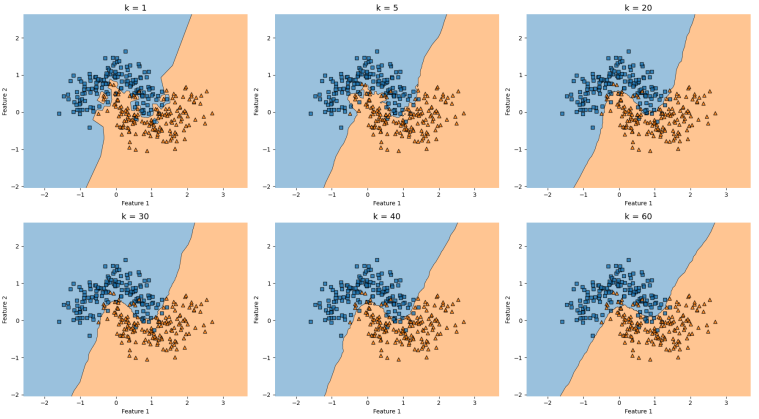
Section 3

WHAT IS K-NEAREST NEIGHBORS (KNN)?

Definition: K-Nearest Neighbors is a simple classification algorithm based on the majority class of the nearest neighbors.

How it Works:

- Calculate the distance between the test image and all training images.
- Find the 'k' nearest neighbors and assign the most common class.



WHY USE KNN

Pros:

- Simple to understand and implement.
- Effective for small datasets.

Cons:

- Slow for large datasets (due to distance calculations).
- Performance depends on the choice of ’k’ and distance metric.

EVALUATION

Confusion Matrix:

	Actually Positive	Actually Negative
Predicted Positive	True Positives (TPs)	False Positives (FPs)
Predicted Negative	False Negatives (FNs)	True Negatives (TNs)

Metrics:

$$Precision = \frac{TP}{TP + FP}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

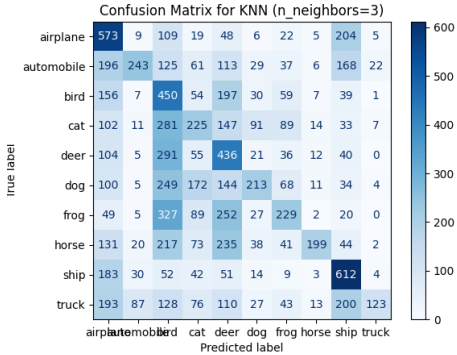
$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precison}$$

KNN ON CIFAR-10

How KNN is used for CIFAR-10:

- **Preprocessing**: reshape the image shapes $(32, 32, 3)$ into $(32 \cdot 32 \cdot 3)$ to get a vector for each image.
- Apply KNN to classify CIFAR-10 images based on pixel values.
- Set the number of neighbors, 'k', and compute the accuracy.



STOP 2: RUN THE KNN SCRIPT

- Run in the **part2_knn.py**
- Modify:
  - k value
- Observe how the confusion matrix and accuracy changes.

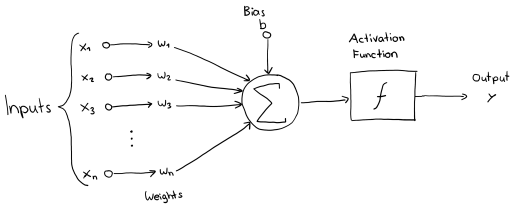
INTRODUCTION INTO NEURAL NETWORKS

Section 4

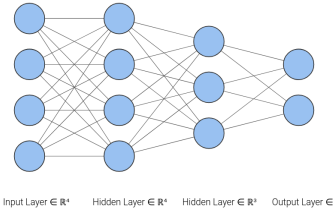
WHAT IS A NEURAL NETWORK

Definition: Neural Networks are a type of deep learning model inspired by the human brain.
Key Components:

- **Neurons** (or units): Basic computational units that process inputs.
- **Layers**: Multiple layers of neurons (input, hidden, output).
- **Activation Functions**: Functions that determine the output of each neuron (e.g., ReLU, Sigmoid).



[1]



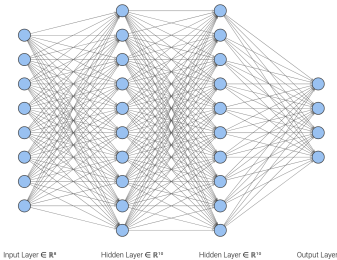
Images generated on https://alexlenail.me/NN-SVG/index.html

HOW NEURAL NETWORKS LEARN

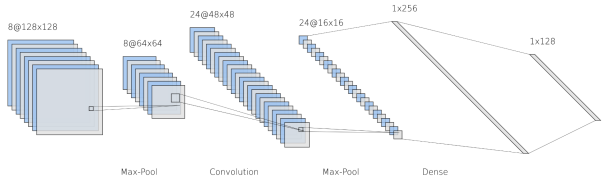
- Neural networks learn by making guesses and checking how wrong they are.
- They adjust internal values (called **weights**) to improve their guesses.
- This process uses:
  - **Backpropagation** – helps figure out what went wrong.
  - **Gradient descent** – decides how much to change the weights.
- **Goal**: Keep improving until the guesses are as accurate as possible.

TYPES OF NEURAL NETWORKS

Fully Connected Neural Networks (FNNs): Basic neural networks with fully connected layers.



Convolutional Neural Networks (CNNs): Special type of NN for image data. They use **convolutional layers** to learn local patterns.



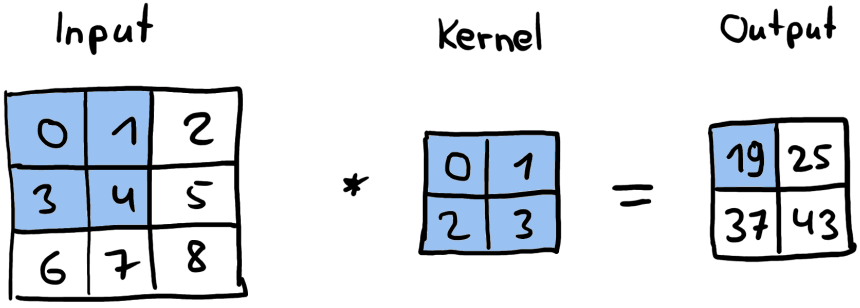
Images generated on `https://alexlenail.me/NN-SVG/index.html`

CONVOLUTIONAL NEURAL NETWORKS (CNNS)

Section 5

WHAT IS CONVOLUTION?

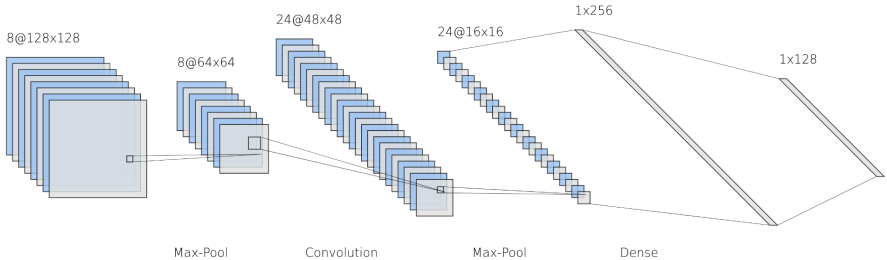
Multiply the **kernel values** by the **original pixel values** of the image and then **sum up the results**.



[2]

WHAT MAKES CNNS DIFFERENT?

- **Convolutional Layers**: Learn spatial hierarchies and local patterns (edges, textures, etc.).
- **Pooling Layers**: Reduce spatial dimensions, making the network more efficient.
- **Fully Connected Layers**: Output the final classification result.



Images generated on `https://alexlenail.me/NN-SVG/index.html`

WHY CNNS PERFORM BETTER THAN KNN

Efficiency in Image Classification:

- CNNs automatically learn hierarchical features from raw images (edges, shapes, objects).
- KNN, on the other hand, requires manually computed features.

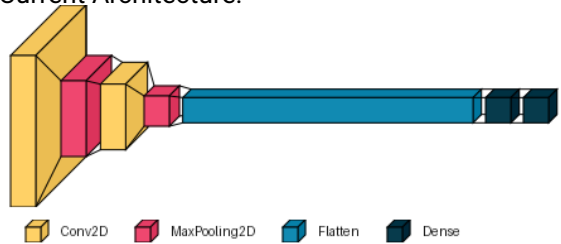
Scalability:

- CNNs perform much better on large, high-dimensional datasets like images.

STOP 3: RUN THE CNN SCRIPT

- Run **part3_cnn.py**.
- Modify:
  - **learning rate**
  - **architecture parameters**
- Observe how the confusion matrix and accuracy changes.

Current Architecture:



BONUS: TRANSFER LEARNING

Section 6

TRANSFER LEARNING

Definition: Reusing a pre-trained model on a new but similar task.
Goal: Leverage powerful models trained on massive datasets (e.g., ImageNet) to classify CIFAR-10.

WHY TRANSFER LEARNING?

Benefits:

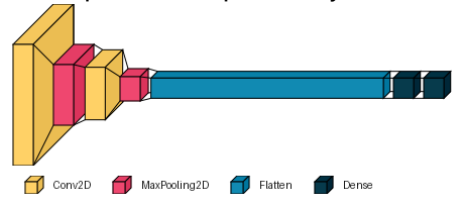
- Works well with **limited data**.
- **Fast training**, often requires fewer resources.
- **Higher accuracy** with minimal effort.

Perfect for laptops and short workshops.

MODEL USED - MOBILENETV2

- Lightweight CNN trained on **ImageNet**.
- Suitable for real-time and mobile applications.
- We only **train a new classification head** on top of it.

In comparison the previously used CNN Architecture:



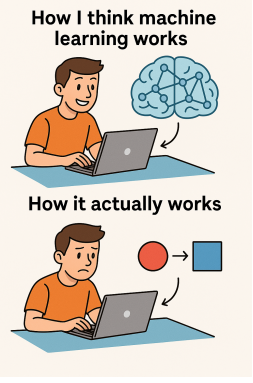
TRANSFER LEARNING WORKFLOW

1. Load pre-trained model (without top layers).
2. Freeze its weights.
3. Add new layers (classifier) for CIFAR-10.
4. Train only the top layers.

STOP 4: RUN THE TRANSFER LEARNING SCRIPT

- Run **part4_transfer_learning.py**.
- Modify:
  - **Learning rate**
  - **Batch size**
  - **Dense layer size**
- Observe how the confusion matrix and accuracy changes.

THANK YOU!



Github: jan1na/Machine-Learning-Workshop

REFERENCES

Section 7

[1] Arthur Arnx. *First neural network for beginners explained (with code)*. Based on this source. Accessed: 2025-04-14. 2019. URL: https://medium.com/data-science/first-neural-network-for-beginners-explained-with-code-4cfd37e06eaf.

[2] Jorgecardete. *Convolutional Neural Networks: A Comprehensive Guide*. Based on this source. Accessed: 2025-04-14. 2024. URL: https://medium.com/thedeephub/convolutional-neural-networks-a-comprehensive-guide-5cc0b5eae175.