Big Data and Automated Content Analysis (12EC)
Week 13: »Multimedia Data«
Wednesday

Damian Trilling
d.c.trilling@uva.nl, @damian0604

May 10, 2023

UvA RM Communication Science

Today

Multimedia data

- The shift to wards multimedia data
- Representing multimedia data
- Classic SML
- Deep Learning
- (Commercial) APIs

Next steps

Today: Beyond words: Multimedia Data

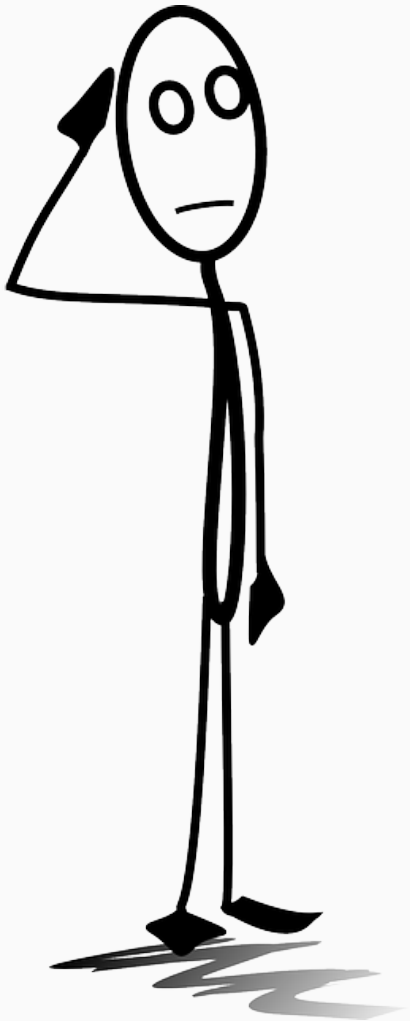
Multimedia data

Multimedia data

The shift to wards multimedia data

An old problem...

- Quantiative content analysis can be applied not only to text, but also to visuals (e.g., Bock et al., 2011)
- Not often done: labor intensive, (easy) data availability, ...
- Yet, more and more important in online settings (e.g., Araujo et al., 2020)



Maybe computational methods can help?

Yes, they can!

A recent turn towards automated content analysis of visual content – incluing a whole special issue on "images as data" in CCR (Casas & Williams, 2022):

1. Images become more important (e.g., Instagram)
2. More images available to social scientists (not only social media, but also repositories, archives, . . .)
3. New computational methods available

Yet, not as mainstream as compuational textual methods

"Moreover, [. . .] there are starting costs to learning [. . .] computer vision methods, given that the jargon is often very computer-science and machine- learning specific, and state-of-the-art libraries and packages are available mostly in Python while social scientists are often more used to working in the R programming language. Thus the special issue aims to lower start-up costs for scholars interested in using images-as-data methods in their research." (Casas & Williams, 2022, p. 3)

And yet, as we'll see, it's not that big a leap from what we already know!

Some examples

Chen et al., 2022

Can we detect conspiracy videos?

It seems we can, based (also) on visual features like contrast and brightness.

Techniques: Manual feature extraction + classical ML

Some examples

Jürgens et al., 2022

How are age and gender represented on German TV?

There seems to be discrimination against women and elderly

Techniques: Deep learning using pre-trained models

Some examples

Joo and Steinert-Threlkeld, 2022

How do politcians represent themselves visually in the media?

It seems there are party and gender differences

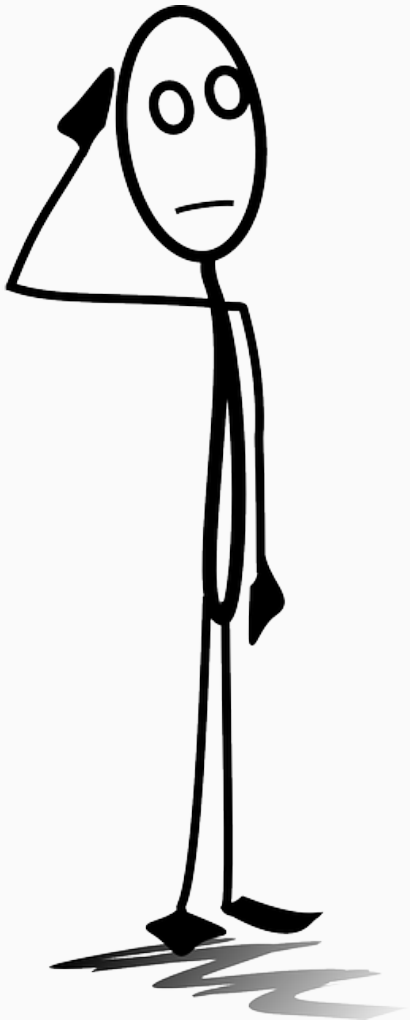
Techniques: Commercial API

Let's have a look at the table of contents of the special issue:
https://computationalcommunication.org/ccr/issue/view/8

Latest trends

- moving images
- combining text and visuals
- really hot: Transformers[1] that model them simultaneously (!!!): "we present data2vec, a framework that uses the same learning method for either speech, NLP or computer vision. The core idea is to predict latent representations of the full input data" (Baevski et al., 2022)

[1] Think: a more fancy thing than embeddings that also takes into account that the same word can have a different meaning in different contexts



But how do we do all of this? (not data2vec, that's for another time...)

Multimedia data

Representing multimedia data

Two types of pictures

Pixel graphics (also called bitmap or raster graphic)

- a matrix of pixels ($\approx$ square dots) of a color
- therefore, loosing quality when scaling (especially up)
- photos, screenshots, ...
- jpg, png, tiff, ...

Two types of pictures

Vector graphics

- geometric shapes
- therefore, fully scalable
- drawings, plots, ...
- svg, eps, ...

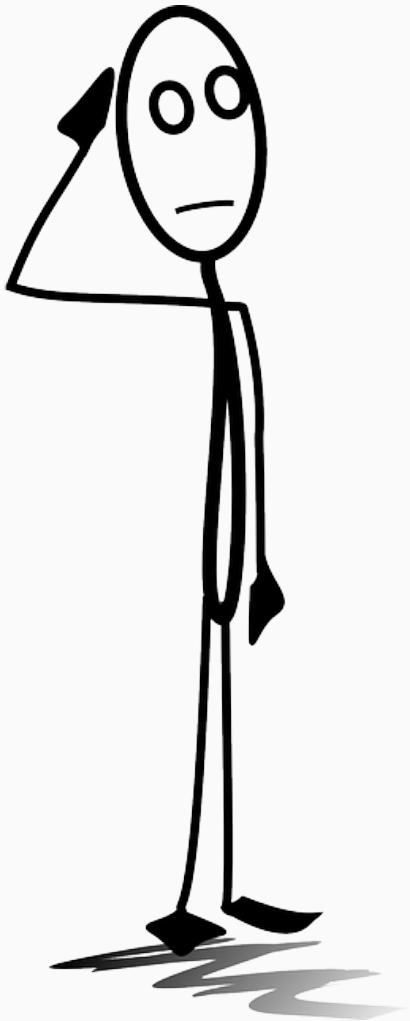
We can easily convert vector graphics to pixel graphics – the other way around only approximately (or not at all)

Two types of pictures

Vector graphics

- geometric shapes
- therefore, fully scalable
- drawings, plots, ...
- svg, eps, ...

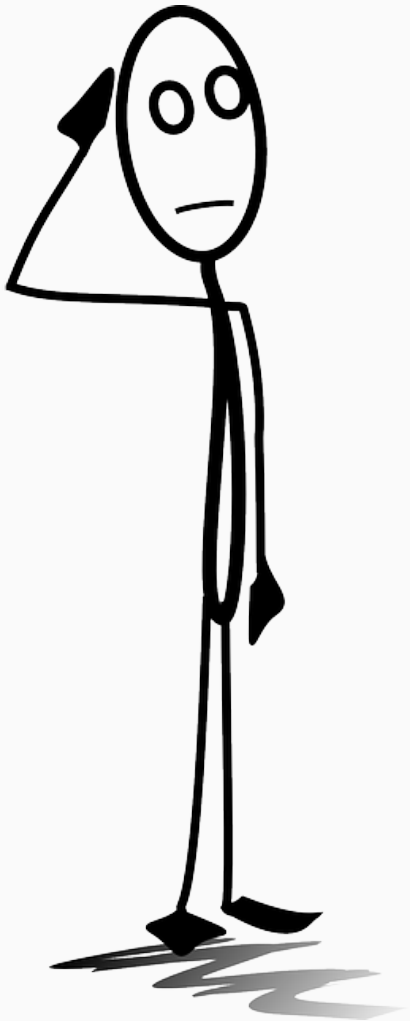
We can easily convert vector graphics to pixel graphics – the other way around only approximately (or not at all)



Not focus of today, but important: How would you store graphs for a report you want to publish?

Our focus today

- We focus on pixel graphics
- Note that moving images are just a series of pixel graphics! (traditional movies: 24 per second)
- Hence, the techniques discussed today can be applied to (stills from) films as well



If pixel graphics are just a series of colored points, do we need something like the vectorizer we needed to transform words to numbers?

Representing pixel graphics

Grayscale images

- If the picture is 200×300 pixel, we have a two-dimensional matrix with $200 \times 300 = 60,000$ pixels
- With a color depth of 1 byte = 8 bit = $2^8 = 256$, each of these pixels is an integer between 0 and 255
- **Two dimensions instead of one (for text)** – but we can *flatten* the matrix to a one-dimensional vector ("vectorize" it, if you want...) (in this case, of length 60,000)

Representing pixel graphics

Color images (RGB)

- Per pixel, we now have three color values (Red, Green, Blue) instead of one (Gray)
- Hence, our matrix becomes three-dimensional:
  $200 \times 300 \times 3 = 180,000$ integers between 0 and 255

What can we do with images in Python?

- PIL (Pillow), the Python Image Library, provides many typical image operations (reading, displaying, cropping, color transformations)
- That's cool for batch processing (a for-loop and you can resize thousands of images...)
- But more imporantly: **It's just a vector/matrix of integers, so we can do machine learning just like we did before!**

Multimedia data

Classic SML

First approach

- If each picture is just a vector of integers, scikit-learn works *exactly the same* as for survey data (Chapter 8) or text (Chapter 11).
- Typical example: recognizing hand-written characters with a Random Forest (Example 14.10) or a Support Vector Machine[2]

[2] https://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html

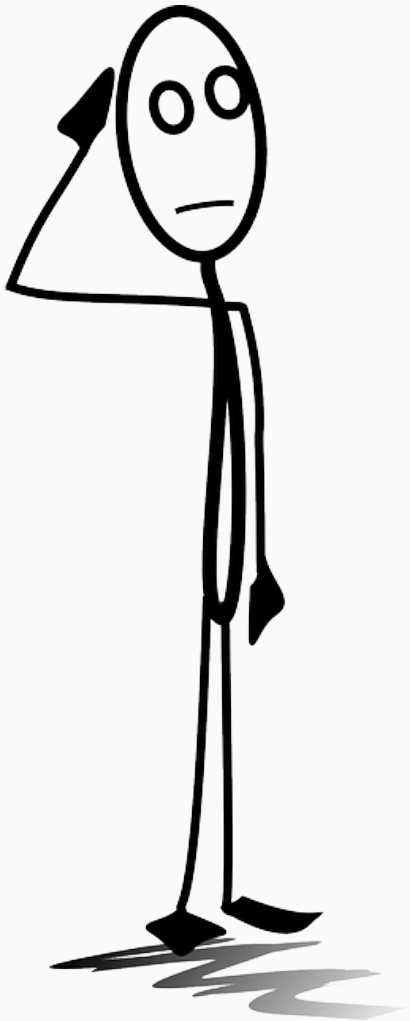
This is impressive!

```
Classification report for classifier SVC(gamma=0.001):
              precision    recall  f1-score   support

           0       1.00      0.99      0.99        88
           1       0.99      0.97      0.98        91
           2       0.99      0.99      0.99        86
           3       0.98      0.87      0.92        91
           4       0.99      0.96      0.97        92
           5       0.95      0.97      0.96        91
           6       0.99      0.99      0.99        91
           7       0.96      0.99      0.97        89
           8       0.94      1.00      0.97        88
           9       0.93      0.98      0.95        92

    accuracy                           0.97       899
   macro avg       0.97      0.97      0.97       899
weighted avg       0.97      0.97      0.97       899
```

https://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html

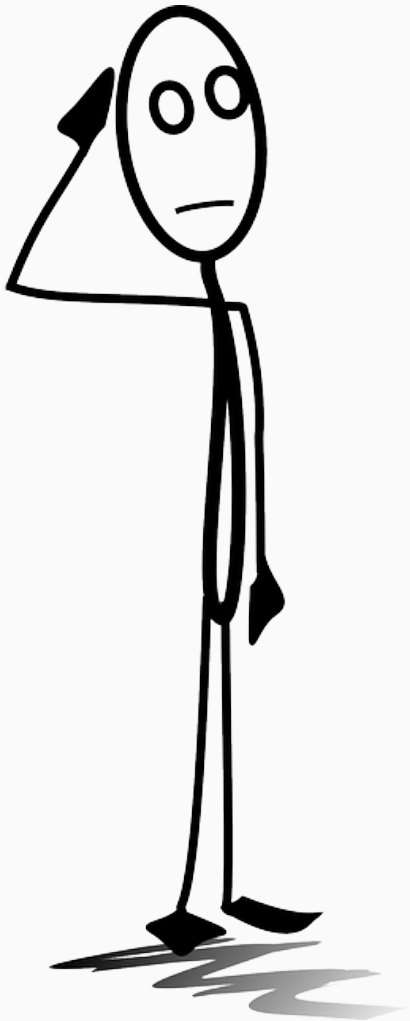


Can you explain why this works?

The intuition behind it

If the images are cropped and resized equally (!)...

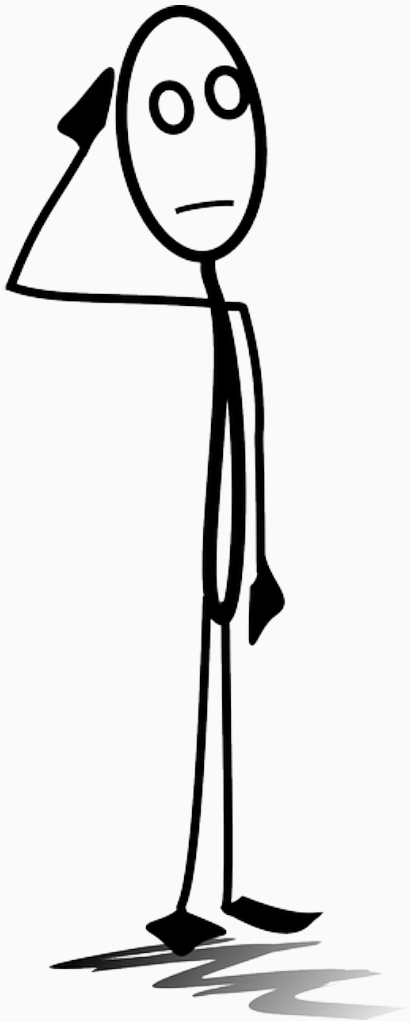
...then the pixels in the center should be white for a 0 but dark for a 8.



Can you name tasks for which you think this would not work?

Multimedia data

Deep Learning



Do you remember what the reason for using deep learning in text classification was?

Say we want to recognize whether an image is a cat or a dog...

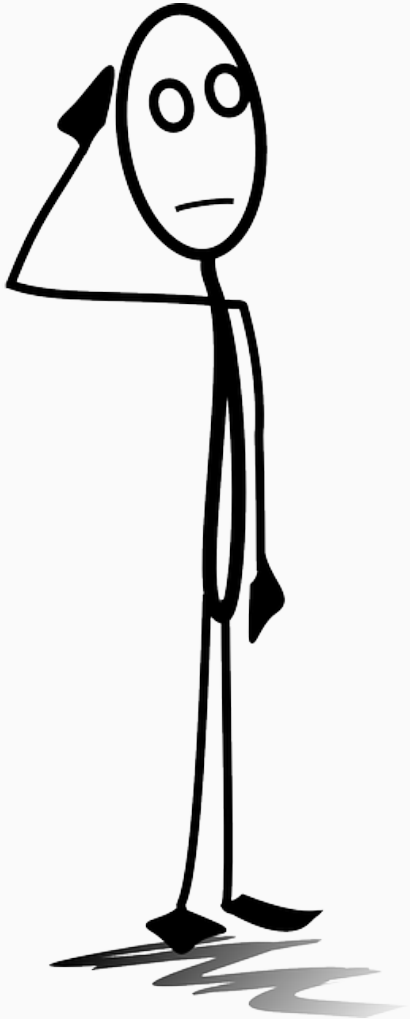
- Can we really say that the color of a pixel maps directly to the "catness" or "dogness" of the picture?
- Or should it not rather be about their fur, the shape of the ears, ...
- But it seems impossible to engineer these features by hand :-(

Let's use a deep neural network to learn them!

Say we want to recognize whether an image is a cat or a dog...

- Can we really say that the color of a pixel maps directly to the "catness" or "dogness" of the picture?
- Or should it not rather be about their fur, the shape of the ears, ...
- But it seems impossible to engineer these features by hand :-(

Let's use a deep neural network to learn them!



Are you really seriously interested in classifying dogs and cats? There are many tutorials, e.g. here:

https://machinelearningmastery.com/

how-to-develop-a-convolutional-neural-network-to-classify-photos-of

How do we do this?

- For deep learning, we use `keras` instead of `scikit-learn`
- You need to specify the *architecture* of the network
- The process is resource-intensive

Different architectures

There are many. . . , but particularly interesting for us:

1. The multilayer perceptron (MLP): A fully-connected feedforward neural network (see Fig. 1)
2. The Convolutional Neural Network (CNN): Layers are not fully connected – think: regularized versions of MLP

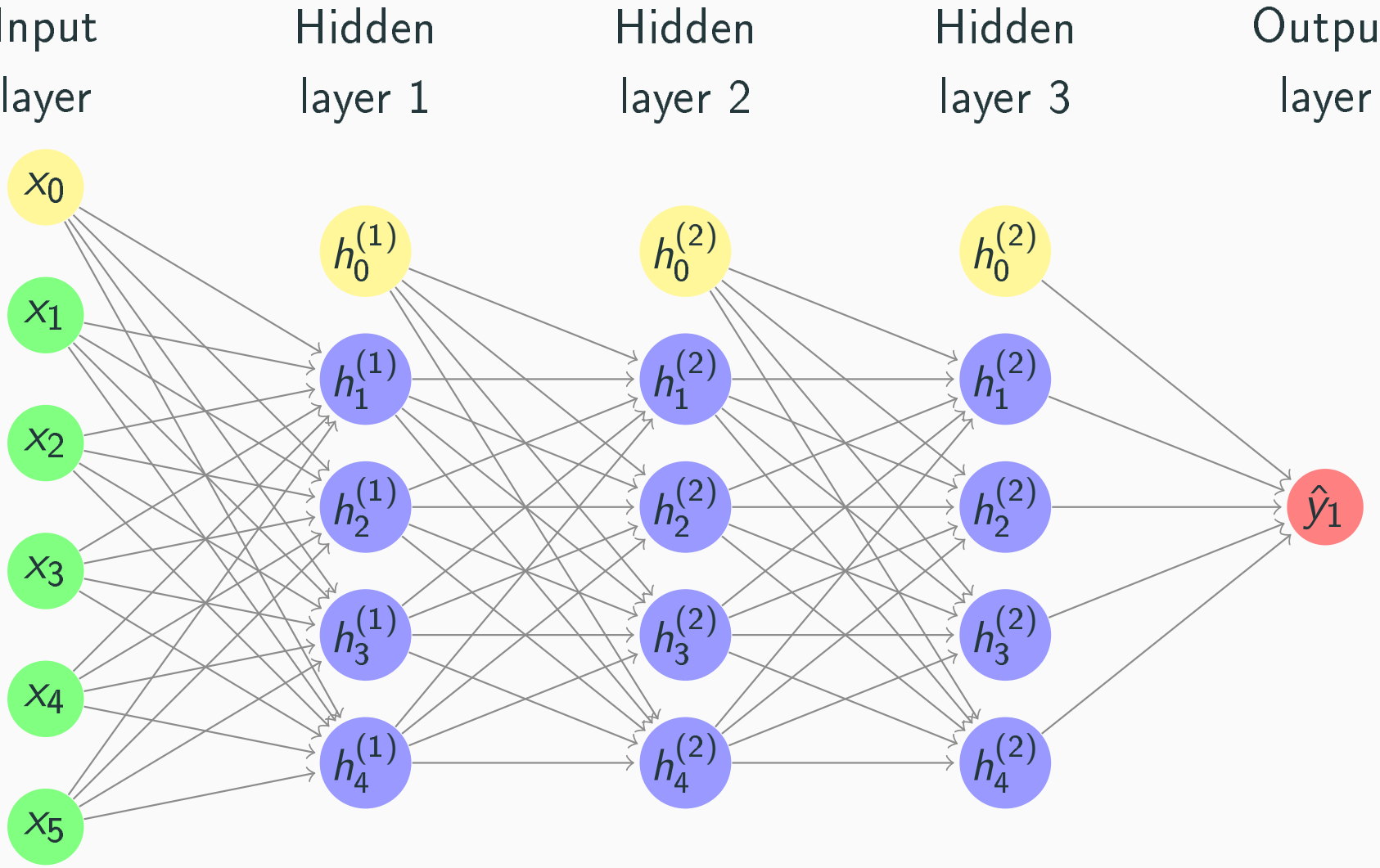


Figure 1: Architecture of a multilayer perceptron for binary predictions. The activation function of the output layer is the (logistic) sigmoid function; the other activation functions can be anything (e.g., ReLu)

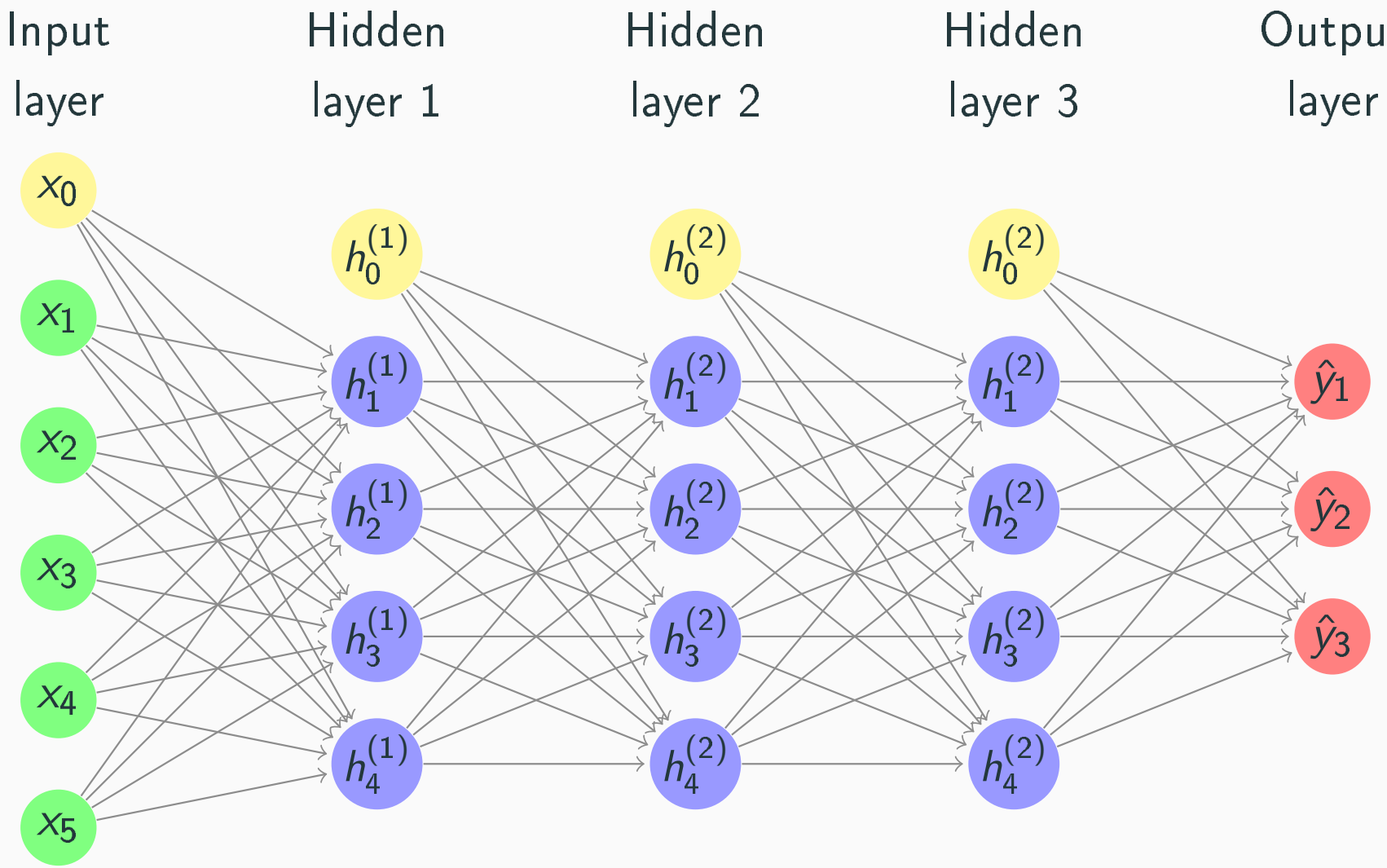


Figure 2: Architecture of a multilayer perceptron for multiclass predictions. The activation function of the output layer is the softmax function (like logistic, but probabilities add up to one); the other activation functions can be anything (e.g., ReLu)

Convolutional Neural Networks

- You start with a convolution layer
- A convolution is a mathematical operation on two functions[3]
- Think of it as a filter (say, sized 5×5 pixels) that slides over the image
- The result goes through a pooling layer that aggregates the outcome (e.g., using the max)
- Optionally: More convolution and pooling layers
- Finally, one or more fully-connected layers

[3] cross-correlation with one of the functions mirrored around the y-axis

Convolutional Neural Networks

- You start with a convolution layer
- A convolution is a mathematical operation on two functions[3]
- Think of it as a filter (say, sized 5×5 pixels) that slides over the image
- The result goes through a pooling layer that aggregates the outcome (e.g., using the max)
- Optionally: More convolution and pooling layers
- Finally, one or more fully-connected layers

[3] cross-correlation with one of the functions mirrored around the y-axis

Convolutional Neural Networks

- You start with a convolution layer
- A convolution is a mathematical operation on two functions[3]
- Think of it as a filter (say, sized 5×5 pixels) that slides over the image
- The result goes through a pooling layer that aggregates the outcome (e.g., using the max)
- Optionally: More convolution and pooling layers
- Finally, one or more fully-connected layers

[3] cross-correlation with one of the functions mirrored around the y-axis

Convolutional Neural Networks

- You start with a convolution layer
- A convolution is a mathematical operation on two functions[3]
- Think of it as a filter (say, sized 5×5 pixels) that slides over the image
- The result goes through a pooling layer that aggregates the outcome (e.g., using the max)
- Optionally: More convolution and pooling layers
- Finally, one or more fully-connected layers

[3] cross-correlation with one of the functions mirrored around the y-axis

Convolutional Neural Networks

- You start with a convolution layer
- A convolution is a mathematical operation on two functions[3]
- Think of it as a filter (say, sized 5×5 pixels) that slides over the image
- The result goes through a pooling layer that aggregates the outcome (e.g., using the max)
- Optionally: More convolution and pooling layers
- Finally, one or more fully-connected layers

[3] cross-correlation with one of the functions mirrored around the y-axis

Convolutional Neural Networks

- You start with a convolution layer
- A convolution is a mathematical operation on two functions[3]
- Think of it as a filter (say, sized 5×5 pixels) that slides over the image
- The result goes through a pooling layer that aggregates the outcome (e.g., using the max)
- Optionally: More convolution and pooling layers
- Finally, one or more fully-connected layers

[3] cross-correlation with one of the functions mirrored around the y-axis

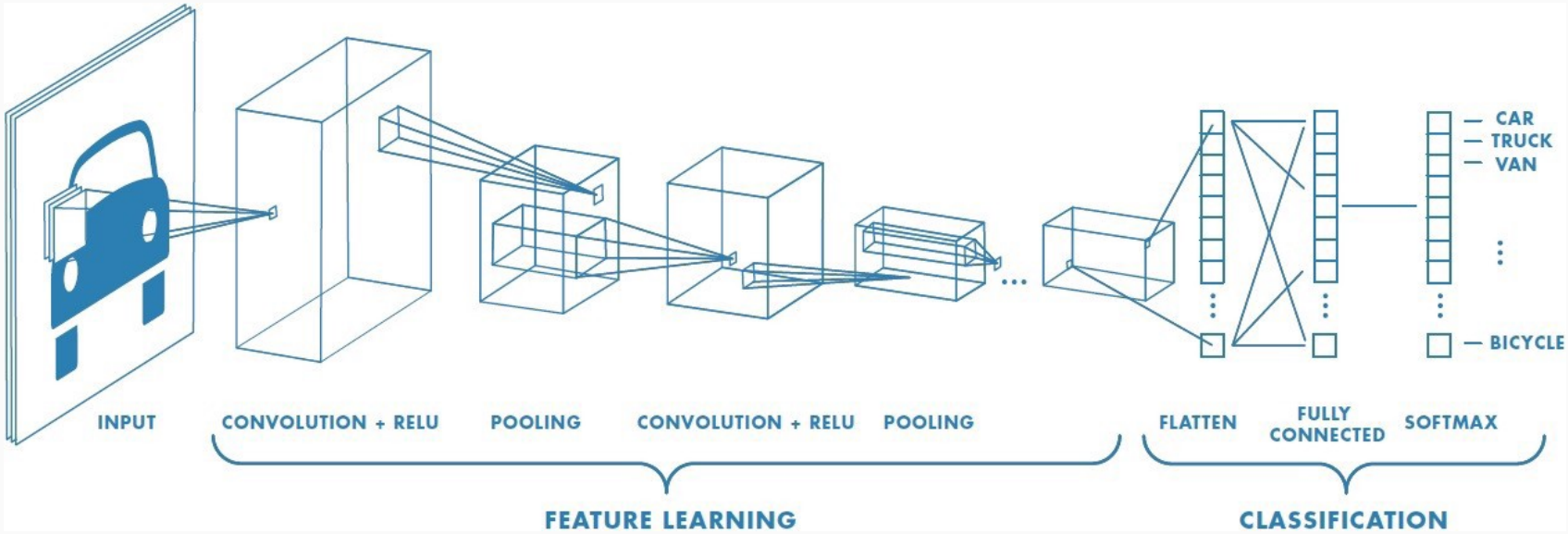


Figure 3: Architecture of a convolutional neural network. Source: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b

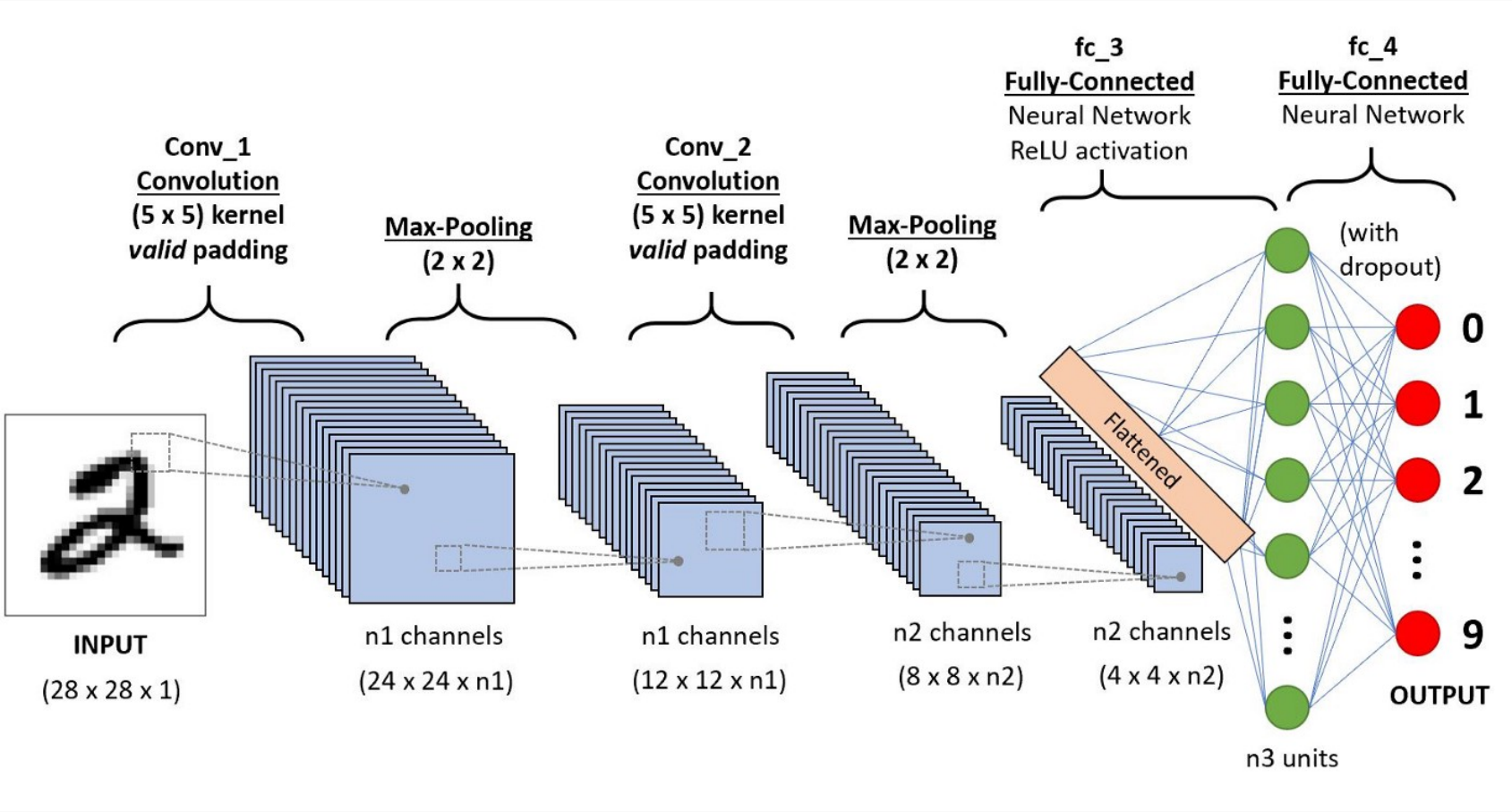


Figure 4: Architecture of a convolutional neural network. Source: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b

Convolutional Neural Networks

- CNNs are more modern than MLPs and
- Because they integrate a *feature learning* step before the *classification* step, they need less preprocessing (Fig. 3
- CNNs are more resource-intensive to train

Convolutional Neural Networks

- CNNs are more modern than MLPs and
- Because they integrate a *feature learning* step before the *classification* step, they need less preprocessing (Fig. 3
- CNNs are more resource-intensive to train

Convolutional Neural Networks

- CNNs are more modern than MLPs and
- Because they integrate a *feature learning* step before the *classification* step, they need less preprocessing (Fig. 3
- CNNs are more resource-intensive to train

Training your own neural network?

If you have a very specific task and an annotated dataset – sure!

But:

- Can be *very* resource-intensive
- Often, very large amounts of training data needed
- There are so many architectures, and you can't possibly know the best ones without being an expert

Training your own neural network?

Solution 0: Just do it ;-)

Training a MLP if not that difficult (Examples 14.11–14.14)

Training your own neural network?

Solution 1: Using a pre-trained model

Just use a CNN that already is trained (Examples 14.14–14.17).

Make sure to read up on the specific model you are using.

Training your own neural network?

Solution 2: Fine-tuning a pre-trained model

If you have a labeled dataset, you can start with an existing pre-trained model and then fine-tune it with your data (many tutorials online)

There is also a really, really cool approach called CLIP that uses transformers to embed images and texts in the same (!) vector space (Radford et al., 2021) – https://huggingface.co/docs/transformers/model_doc/clip

Training your own neural network?

Solution 3: A (commercial) API

(next section)

Multimedia data

(Commercial) APIs

What is it?

- You send an image to the API and get a set of labels back
- Under the hood: pre-trained neural networks
- Very much black box: you have no idea where the labels come from
- Prominent players: Clarifai, Google Cloud Vision, Microsoft
- Usually paid but often free for small projects and/or academic use

Automated Visual Content Analysis (Araujo et al., 2020)

Problem

The labels do not correspond what one may be theoretically interested in.

Solution

- Annotate a set of images manually using a codebook
- Use the labels provided by the API as features to train a classifier to predict the annotations
- Thus, we essentially have transformed the image prediction task to a textual prediction task

Automated Visual Content Analysis (Araujo et al., 2020)

Problem

The labels do not correspond what one may be theoretically interested in.

Solution

- Annotate a set of images manually using a codebook
- Use the labels provided by the API as features to train a classifier to predict the annotations
- Thus, we essentially have transformed the image prediction task to a textual prediction task

Is it difficult to use a computer vision API?

No.

All providers provide Python code examples, e.g. https://cloud.google.com/vision/docs/detect-labels-image-client-libraries

Further reading

- all the articles in the special issue edited by Casas and Williams, 2022
- a short book by Webb Williams et al., 2020
- the study by Araujo et al., 2020
- This medium post on CNNs: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

Final remark

The black box nature of (most of) these techniques *is* a big problem from a social-science and epistemological point of view: in particular, known and un-known biases!

As always, but maybe especially here: Never take the outcome as a given – it needs interpretation and critical reflection!

Next steps

Let's have a look together at the course manual regarding the final project.

Examples

On Canvas under “modules”, you find some example projects. Note that these are not necessarily *perfect* assignments, and also that the content of the course (and hence, expectations) are not be fully identical to the years in which a specific assignment was made. **In particular, there may be more advanced and more appropriate techniques available now!**

Friday

On Friday, you can *either* work on one of the computer vision examples discussed today and/or in the book.

Or you start working on your final project.

References

Araujo, T., Lock, I., & van de Velde, B. (2020). Automated visual content analysis (AVCA) in communication research: A protocol for large scale image classification with pre-trained computer vision models. *Communication Methods and Measures*, *14*(4), 239–265. https://doi.org/10.1080/19312458.2020.1810648

Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., & Auli, M. (2022). data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language.arXiv 2202.03555. http://arxiv.org/abs/2202.03555

Bock, A., Isermann, H., & Knieper, T. (2011). Quantitative content analysis of the visual. *The SAGE handbook of visual research methods*, 265–282.

Casas, A., & Williams, N. W. (2022). Introduction to the special issue on images as data. *Computational Communication Research*, *4*(1). https://doi.org/10.5117/ccr2022.1.000.casa

Chen, K., Kim, S. J., Gao, Q., & Raschka, S. (2022). Visual framing of science conspiracy videos. *Computational Communication Research*, *4*(1), 98–134. https://doi.org/10.5117/ccr2022.1.003.chen

Joo, J., & Steinert-Threlkeld, Z. C. (2022). Image as data: Automated content analysis for visual presentations of political actors and events. *Computational Communication Research*, *4*(1), 11–67. https://doi.org/10.5117/ccr2022.1.001.joo

Jürgens, P., Meltzer, C. E., & Scharkow, M. (2022). Visual framing of science conspiracy videos. *Computational Communication Research*, *4*(1), 173–207. https://doi.org/10.5117/ccr2022.1.005.jurg

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision (M. Meila & T. Zhang, Eds.). In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning*, PMLR. https://proceedings.mlr.press/v139/radford21a.html

Webb Williams, N., Casas, A., & Wilkerson, J. D. (2020). *Images as data for social science research: An introduction to convolutional neural nets for image classification*. Cambridge University Press. https://doi.org/10.1017/9781108860741