Machine Learning (ML) for Data-Driven Decision-Making (DDDM)

Module 4: Deep Learning

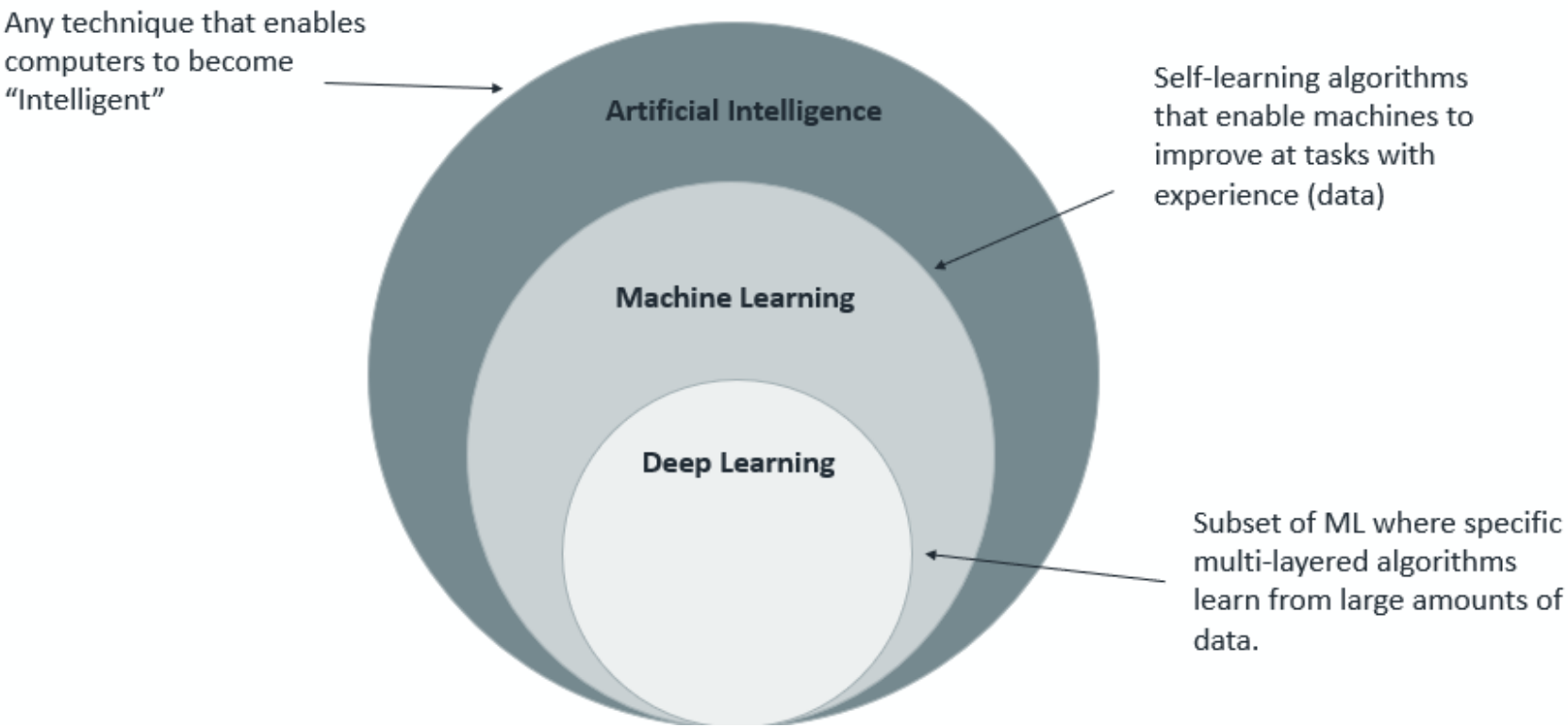
Tobias Rebholz

University of Tübingen

Summer Term 2024

Deep Learning

Deep Learning



(https://towardsdatascience.com/redefining-data-science-d7f2026a021a)

Deep Learning

- Deep Learning (DL) is a sub-field of Machine Learning that uses multi-layered or "deep" neural networks (NNs) for classification, regression, and other DDDM tasks
- Motivation: Most of the world's data is held in unstructured formats
  - NNs can process unstructured data (e.g., text and images) quite well
  - Classical ML algorithms usually leverage structured data to make predictions
- NNs automate feature extraction, removing some of the dependency on human experts
  - E.g., in a set of photos of different people, DL algorithms can determine which features (e.g., shape of mouth, eyes, ...) are most important in distinguishing emotional expressions from one another
  - In classical ML, this hierarchy of features must be established manually by a human expert
- A DL algorithm adjusts and fits itself (through gradient descent and backpropagation; not discussed!) to improve its out-of-sample prediction performance

Deep Learning

- Good performance of DL algorithms usually requires intensive knowledge, time, and (computational) resources to train the models
  - There is a whole industry trying to make money out of training good DL models
  - In other words, we cannot afford these resources!
- Instead of training the models ourselves, we focus on using pre-trained models to solve our own DDDM problems in this module
  - Costs: We give up control over what the (pre-)trained model can actually do—and some understanding of how it achieved this competence
- E.g., the Hugging Face community provides access to many pre-trained models and datasets: https://huggingface.co/

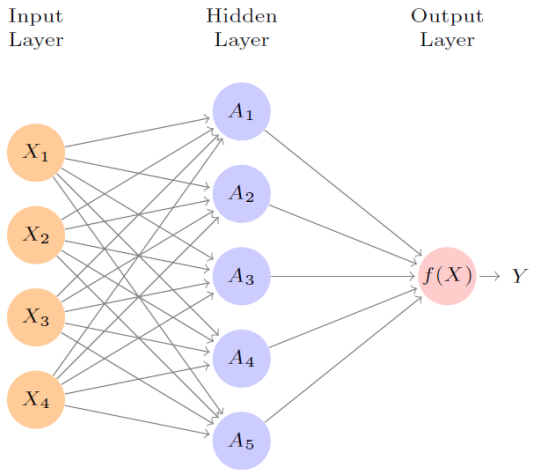
Neural Networks

Neural Networks

- NNs represent state-of-the-art DL technology
  - They are trained on massive amounts of data to identify and classify phenomena, recognize patterns and relationships, evaluate possibilities, make predictions and decisions, ...
- NNs required a lot of tinkering, whereas newer methods (e.g., SVM, RF) are more automatic
  - On many problems, the newer methods outperform **poorly-trained** NNs
  - But: NNs are successful especially on some **niche problems**, such as:
    - Image recognition (first half of today's session)
    - Video classification
    - Speech and text modeling (second half of today's session)
    - ...

Neural Networks

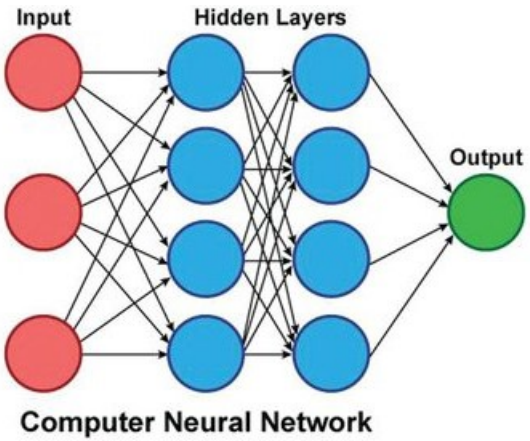
- Single layer NN: The hidden layer computes so-called "activations" $A_k = h_k(X)$ that are nonlinear transformations of linear combinations of the vector of input features $X = (x_1, x_2, \ldots, x_p)$
  - "Hidden" because the activations A_k are not directly observed
  - The functions $h_k(\cdot)$ for transforming the observed features are not fixed in advance, but learned during the training of the NN
  - The output layer is a linear model that uses the activations A_k as inputs, resulting in a function $f(X)$ to make predictions (cf. regression)



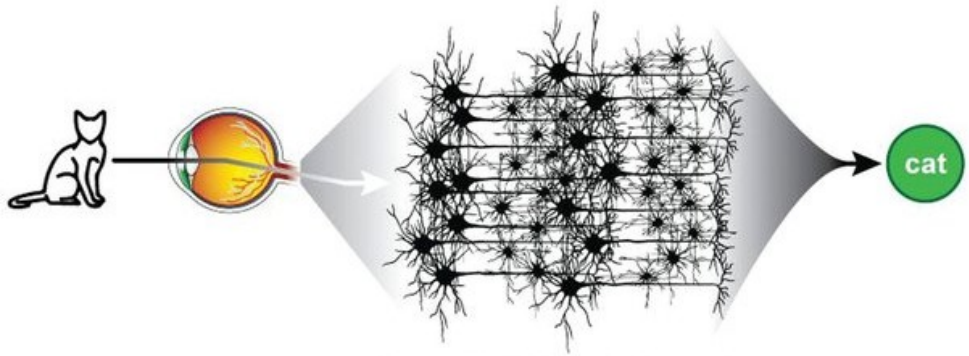
(James et al., 2021, Figure 10.1)

Neural Networks

- NNs are inspired by the functioning of the human brain:
  - Activations A_k: Neurons receive signals (inputs) from other neurons, process them, and send signals to other neurons
  - Learning of $h_k(\cdot)$: This process is akin to the brain's ability to form new connections and strengthen existing ones based on experiences
  - Output: The brain processes information from neurons (i.e., activations) and generates responses (i.e., predictions)



Computer Neural Network

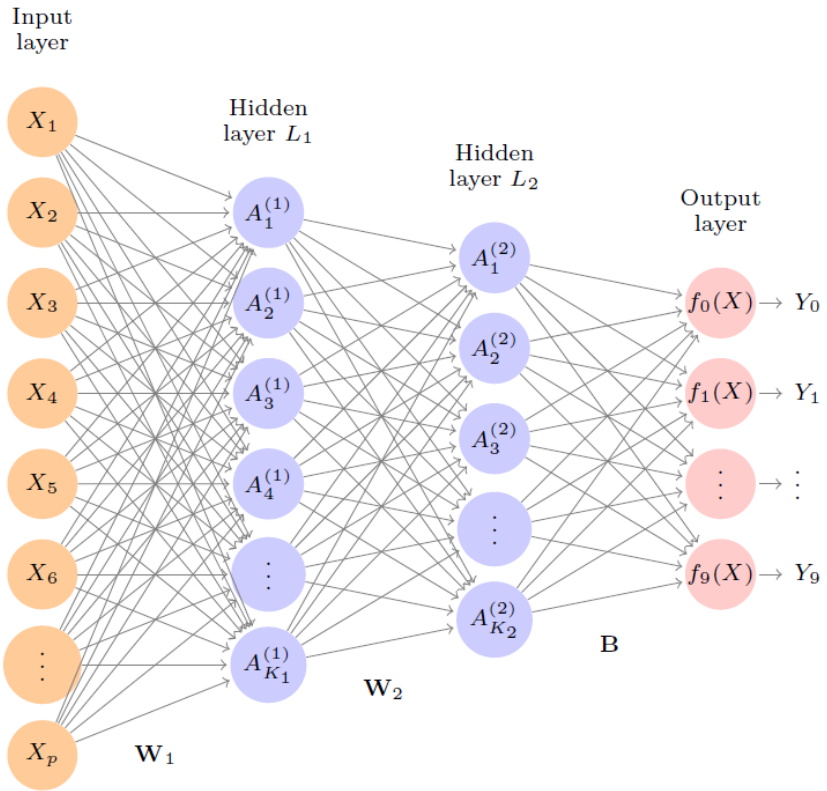


Biological Neural Network

(adapted from https://www.quora.com/How-is-AI-similar-different-to-the-human-brain)

Neural Networks

- More complex NN with multiple hidden layers and outputs:



(James et al., 2021, Figure 10.4)

Most Important Types of NNs

- Feedforward NNs: Simplest type of NN, where information travels in one direction only, from the input layer, through any hidden layers, to the output layer
  - Widely used in pattern recognition and standard classification tasks
- Convolutional NNs: Designed to automatically and adaptively learn spatial hierarchies of features from the input (our focus today!)
  - Primarily used for image processing, classification, and segmentation
- Recurrent NNs: Have "memory" in the form of hidden state vectors that capture information about what has been computed so far
  - Mainly used for sequential data tasks, such as language modeling and time series prediction (more on this later!)
- Autoencoders (AEs): Encoding the input into a latent-space representation, and then decoding the latent representation back to the original input
  - Great for feature extraction, dimensionality reduction, compression, ...

Important Programming Notes

- DL is much more established in Python than in R
  - Therefore, many R packages actually implement Python code in the background
  - Implication: DL tasks that are easy to implement in Python can be quite tricky to get running in R
- `mlr3keras`: The DL package from the `mlr3verse` is still in a very early stage and under active development
  - Therefore, we will use the `reticulate` R-package as an interface to Python in order to access state-of-the-art DL technology (e.g., `tf-keras` for NNs, `transformers` for NLP)

Important Programming Notes

- Preparation in R:
  - This setup chunk only needs to be run once and takes a while until finished

```
if (!('reticulate' %in% installed.packages())) {
  # Python:
  install.packages("reticulate")
  library(reticulate)
  install_miniconda()

  # NNs:
  reticulate::py_install('torch', pip=T)
  reticulate::py_install('tf-keras', pip=T)

  # Transformers:
  reticulate::py_install('transformers', pip=T)

  # Hugging Face:
  reticulate::py_install('datasets', pip=T)
}
```

Image Recognition

Image Recognition

- Image recognition: A classification task where the goal is to assign a label to an image
  - Feature **matrix**: The pixel values that make up an image (incl. color, if applicable)
  - Target: The object shown in the image
    - Note: The target "object" can also be an entire scenery, specific situations or actions, ...
- Performance measurement: Usual metrics (i.e., classification error/accuracy)

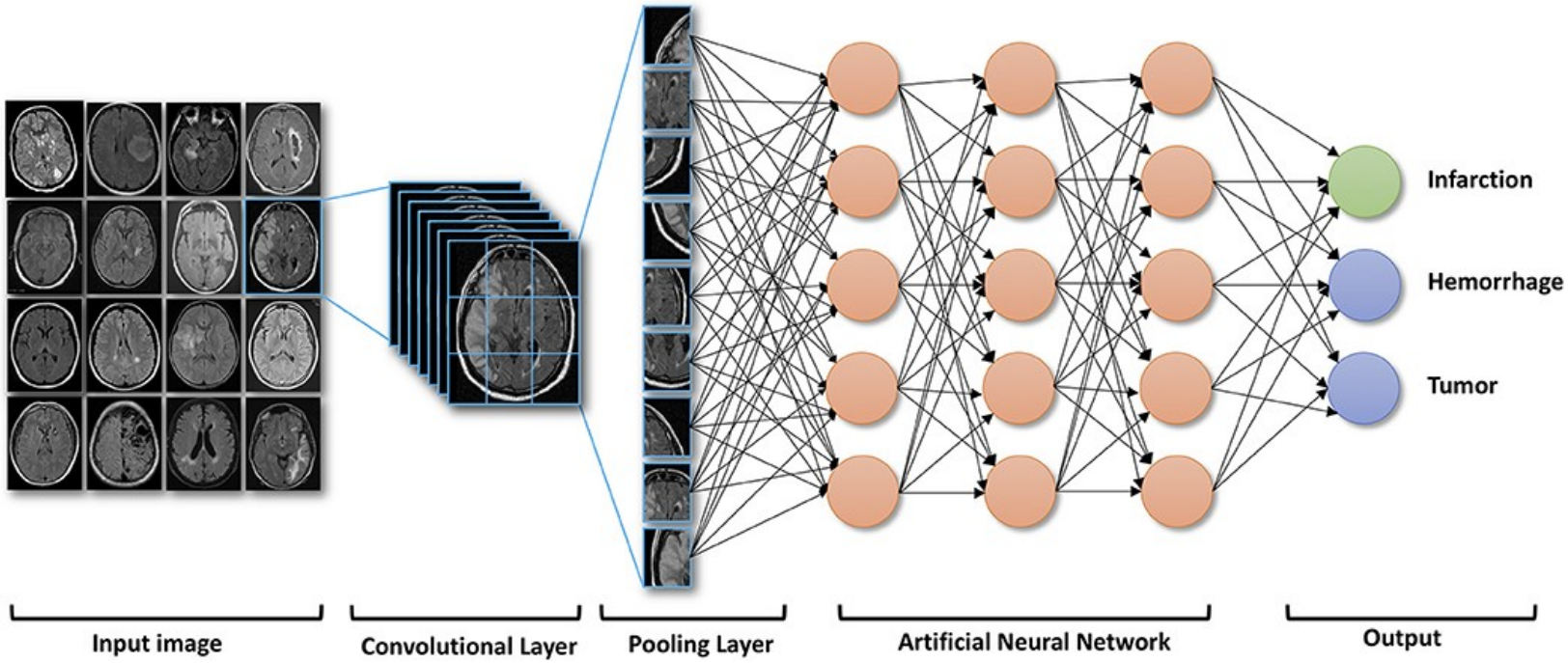
Applications in Behavioral Science: Handwriting Detection

- Classification of handwritten digits:

(James et al., 2021, Figure 10.3)

Applications in Neuroscience: Brain Scans

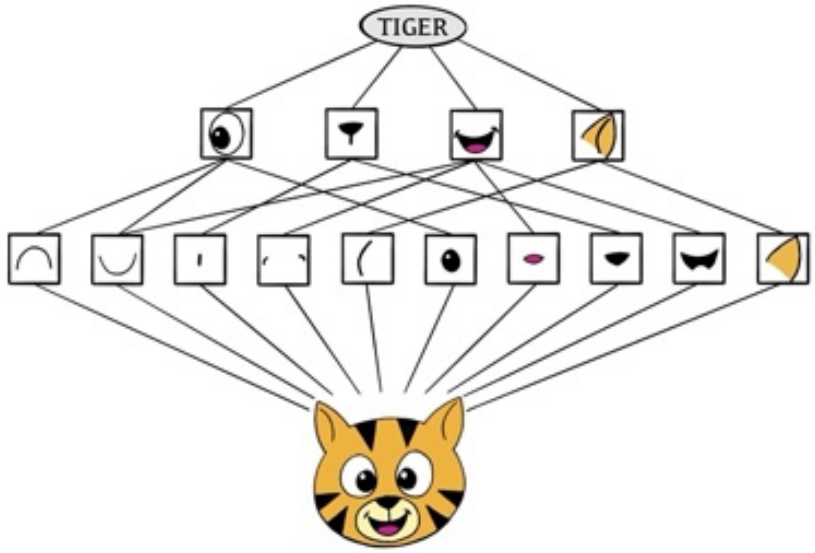
- Screening fMRI scans for signs of disease such as cancer:



(adapted from Zhu et al., 2019)

CNNs for Image Recognition

- CNN layers for image identification:
  - Takes in the image and identifies local features (e.g., a specific shape of a line or a color)
  - Combines the local features in order to create compound features (e.g., eyes and ears)
  - Compound features are used to output the target label "tiger"



(James et al., 2021, Figure 10.6)

Image Recognition in `tf-keras`

- Facial emotion recognition task:

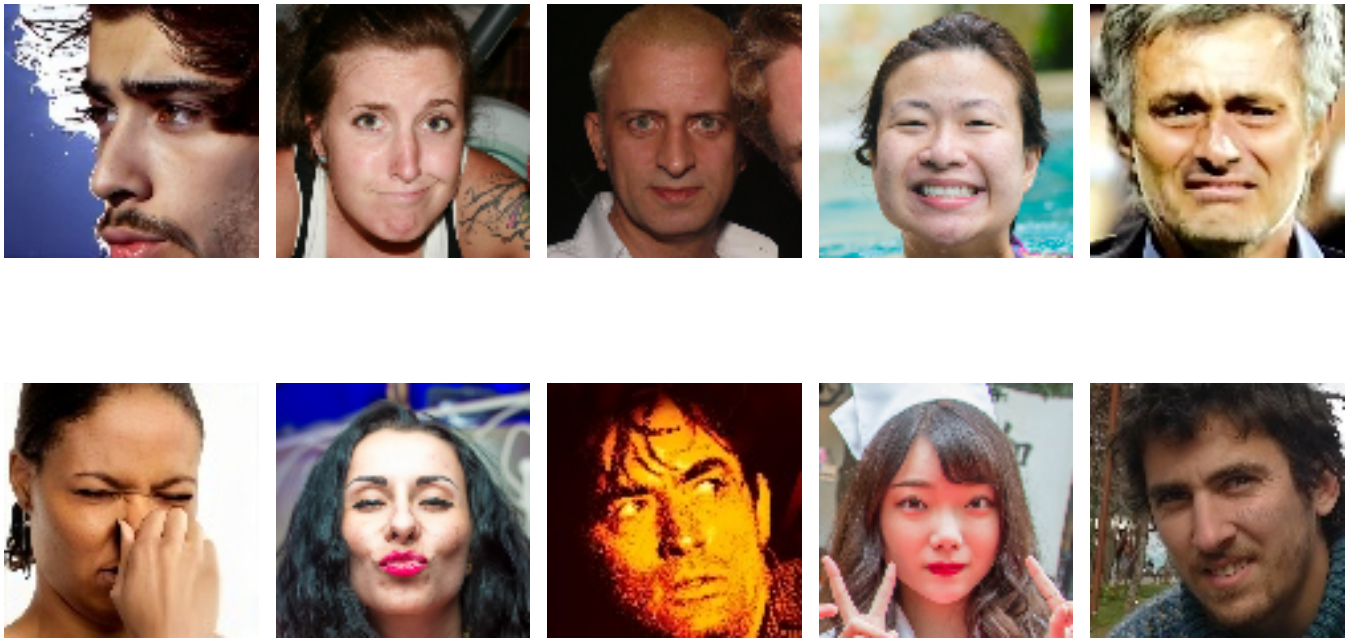


Image Recognition in tf-keras

- Reading the images as data into R:

```
library(reticulate) #; py_config() #; use_condaenv("r-reticulate")
datasets <- import("datasets") #import Python package as "variable" into R
dat = datasets$load_dataset("FastJobs/Visual_Emotional_Analysis", split = 'train')

# Translate true numeric class to verbal label and transforming Python to R data:
dict_emo <- dat$features$label$names
dat <- dat$to_pandas() %>%
  as_tibble() %>%
  mutate(emotion = dict_emo[label+1])

# Selecting a random subset of images:
set.seed(1)
dat_subset <- dat %>% sample_n(10)
head(dat_subset)
## # A tibble: 6 x 3
##   image             label emotion
##   <list>            <dbl> <chr>
## 1 <named list [2]>      6 sad
## 2 <named list [2]>      1 contempt
## 3 <named list [2]>      5 neutral
## 4 <named list [2]>      4 happy
## 5 <named list [2]>      2 disgust
## 6 <named list [2]>      2 disgust
```

Image Recognition in tf-keras

- Loading a pre-trained CNN (as a pipeline, i.e., incl. image data pre-processor) that is fine-tuned for facial emotion recognition:

```
transformers <- import("transformers") #import Python package as "variable" into R
prc <- transformers$AutoImageProcessor$from_pretrained("dennisjooo/emotion_classification")
ppl <- transformers$pipeline(model = 'dennisjooo/emotion_classification',
                             image_processor = prc)
ppl
## <transformers.pipelines.image_classification.ImageClassificationPipeline object at 0x000
##  signature: (
##    images: Union[str, List[str], ForwardRef('Image.Image'), List[ForwardRef('Image.Image
##    **kwargs
## )
```

Image Recognition in tf-keras

- Model predictions:

```
# Exemplary prediction for first image:
ppl(dat_subset$image[[1]]$path) %>% as.data.frame()
##   label     score label.1   score.1 label.2    score.2 label.3    score.3
## 1   sad 0.6794345   anger 0.1221533 disgust 0.05566198    fear 0.05533549
##    label.4    score.4
## 1 contempt 0.02939186

# Predictions for all images:
dat_subset <- dat_subset %>%
  rowwise() %>%
  mutate(pred = ppl(image$path) %>% as.data.frame())
tail(dat_subset)
## # A tibble: 6 x 4
## # Rowwise:
##   image            label emotion  pred$label $score $label.1 $score.1 $label.2
##   <list>           <dbl> <chr>    <chr>       <dbl> <chr>       <dbl> <chr>
## 1 <named list [2]>     2 disgust  disgust     0.360 anger       0.350  sad
## 2 <named list [2]>     2 disgust  disgust     0.438 anger       0.324  sad
## 3 <named list [2]>     1 contempt contempt    0.688 disgust     0.102  happy
## 4 <named list [2]>     3 fear     fear        0.708 anger       0.0705 sad
## 5 <named list [2]>     5 neutral  neutral     0.733 happy       0.0608 surprise
## 6 <named list [2]>     2 disgust  disgust     0.519 anger       0.191  contempt
## # i 5 more variables: pred$score.2 <dbl>, $label.3 <chr>, $score.3 <dbl>,
## #   $label.4 <chr>, $score.4 <dbl>
```

Excurse: Image Generation

- Technically, CNNs cannot only be used for image recognition, but also for image creation
- Generative Adversarial Network (GAN): This is like a game between two players, one player being the Generator (i.e., the artist) and the other player being the Discriminator (i.e., an art critic, who tries to tell if an image is real or fake)
  - The Generator-CNN takes random noise as input and transforms it through several layers to produce an image
    - It starts with basic features and gradually adds more detail to generate a complete image
  - The Discriminator-CNN takes an image from the Generator-CNN as input and processes it through several layers to produce a single output: fake vs. real

Hands-on Practical Tutorial

- Now it's your turn:
  - Go to ILIAS
  - Navigate to the "Tutorials" folder
  - Download the "Module 4" folder
  - Work on the tutorial "module4-DL"

Natural Language Processing

Text Mining Applications

- Classification of news stories, social media posts, journal entries from experience sampling studies, ... according to their content
- Content filtering (e.g., spam emails, hate speech on social media)
- Content summarizing, paraphrasing, and translation, ...
- Clustering of documents and web pages according to shared features (e.g., similar topics)
- Insights about trends (e.g., trending hashtags on Twitter/X)

Challenges in Text Mining

- Large and unstructured textual data bases
  - Sometimes documents are not even available in electronic form
- Very high number of possible "dimensions" (but sparse): All possible word and phrase types in a language
- Complex and subtle relationships between concepts in text, such as:
  - Word ambiguity: E.g., Apple (the company) vs. apple (the fruit)
  - Context sensitivity: E.g., humor
  - Irony, sarcasm, ...
- Rather noisy data: E.g., spelling or grammar mistakes (esp. relevant and problematic for social media data)
- ...

Tokenization

- Tokenization: The process of breaking text into pieces such as words, keywords, phrases, symbols, and other elements called "tokens"
  - Tokens: Often individual words, but they can also be be special characters such as punctuation marks or emojis
  - The tokens from tokenization are machine readable and can then be analyzed using NLP

Tokenization: Most Simple Forms

```
"Psychology is the science of the behavior."
## [1] "Psychology is the science of the behavior."
```

- Bag-of-words:

```
cbind(c("psychology", 1), c("is", 1), c("the", 2), c("science", 1), c("of", 1),
      c("behavior", 1))
##      [,1]         [,2] [,3]  [,4]      [,5] [,6]
## [1,] "psychology" "is" "the" "science" "of" "behavior"
## [2,] "1"          "1"  "2"   "1"       "1"  "1"
```

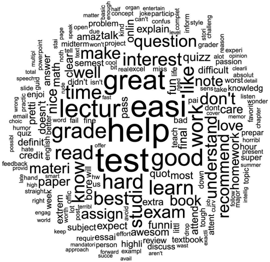
- String-of-words:

```
cbind(c("psychology", 1), c("is", 2), c("the", 3), c("science", 4), c("of", 5),
      c("the", 6), c("behavior", 7))
##      [,1]         [,2] [,3]  [,4]      [,5] [,6]  [,7]
## [1,] "psychology" "is" "the" "science" "of" "the" "behavior"
## [2,] "1"          "2"  "3"   "4"       "5"  "6"   "7"
```

- Why is this distinction so important?
  - Shakespeare's plays contain ca. 885,000 words, but the count of unique word forms is only 29,000 (https://www.opensourceshakespeare.org/statistics/)

Visualizations of Text Data

- Word clouds: Visualizing the information contained in text data (i.e., open comments in teaching evaluations) by displaying the most frequently occurring words
  - The size of each word represents its frequency (as a proxy for its importance) within the text corpus
  - Linking words (e.g., "and"), articles (e.g., "the") and so on are typically excluded because they have no meaning on their own



(Jacobucci et al., Figure 11.10)

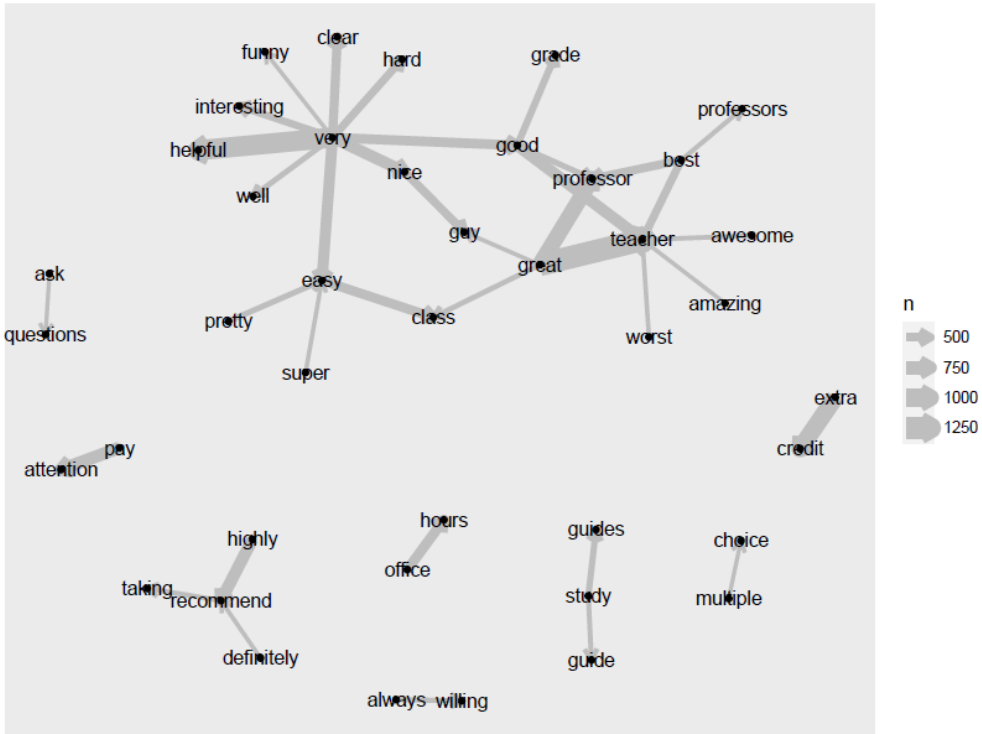
Visualizations of Text Data

- A little more informative in terms of content than individual word clouds: The most frequently occurring bi- (left) and trigrams (right):

(Jacobucci et al., Figures 11.12 & 11.13)

Word Embeddings

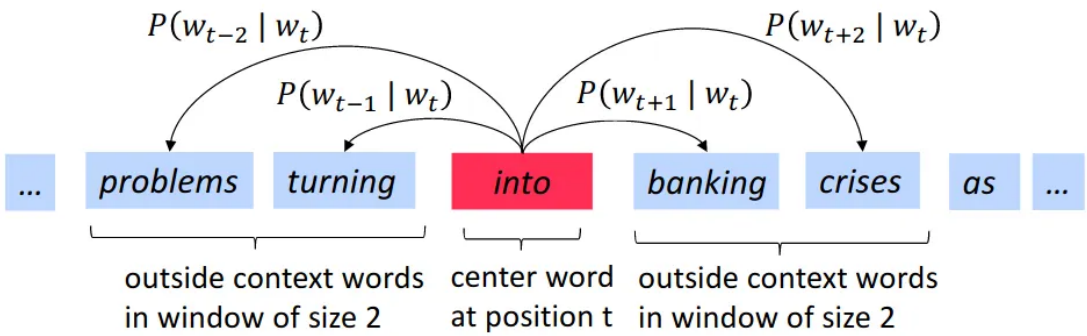
- In general, language is much more than a loose collection of its constituent components
  - Network plot: Visualizing the associations between the top used words:



(Jacobucci et al., Figure 11.14)

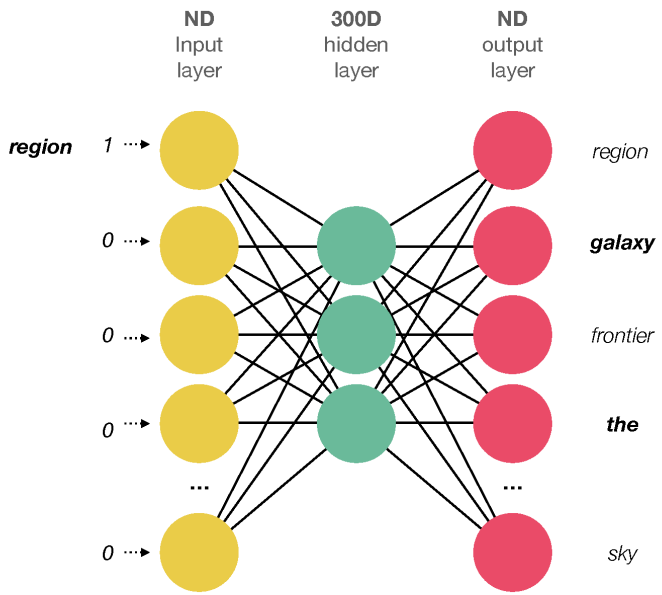
Word Embeddings

- Word2Vec (Mikolov et al., 2013): Unsupervised learning technique to learn continuous, multi-dimensional vector representations of words that capture information about its meaning based on the surrounding words
  - In other words, Word2Vec by design captures the similarity of words in a training corpus (i.e., how far or near a word is in vector space to other words), which represents a notion of word-sense
- Approach: Given a center word, learning to predict the most likely words in a fixed-sized window around it



(https://towardsdatascience.com/word2vec-to-transformers-caf5a3daa08a)

Example



"You're on your way, Kelvin. Good luck!" Moddard's voice sounded as close as before.

A wide slit opened at eye-level, and I could see the stars. The _Prometheus_ was orbiting in the region of Alpha in Aquarius and I tried in vain to orient myself; a glittering dust filled my porthole. I could not recognize a single constellation; in this region of the galaxy the sky was unfamiliar to me. I waited for the moment when I would pass near the first distinct star, but I was unable to isolate any one of them. Their brightness was fading; they receded, merging into a vague, purplish glimmer, the sole indication of the distance I had already travelled. My body rigid, sealed in its pneumatic envelope, I was knifing through space with the impression of standing still in the void, my only distraction the steadily mounting heat.

Suddenly, there was a shrill, grating sound, like a steel blade being drawn across a sheet of wet glass. This was it, the descent. If I had not seen the figures racing across the dial, I would not have noticed the change in direction. The stars having vanished long since, my gaze was swallowed up on the pale reddish glow of infinity. I could hear my heart thudding heavily. I could feel the coolness from the air-conditioning on my neck, although my face seemed to be on fire. I regretted not having caught a glimpse of the _Prometheus_, but the ship must have been out of sight by the time the automatic controls had raised the shutter of my porthole.

The capsule was shaken by a sudden jolt, then another. The whole vehicle began to vibrate. Filtered through the insulating layers of the outer skins, penetrating my pneumatic cocoon, the vibration reached me, and ran through my entire body. The image of the dial shivered and multiplied, and its phosphorescence spread out in all directions. I felt no fear. I had not undertaken this long voyage only to overshoot my target!

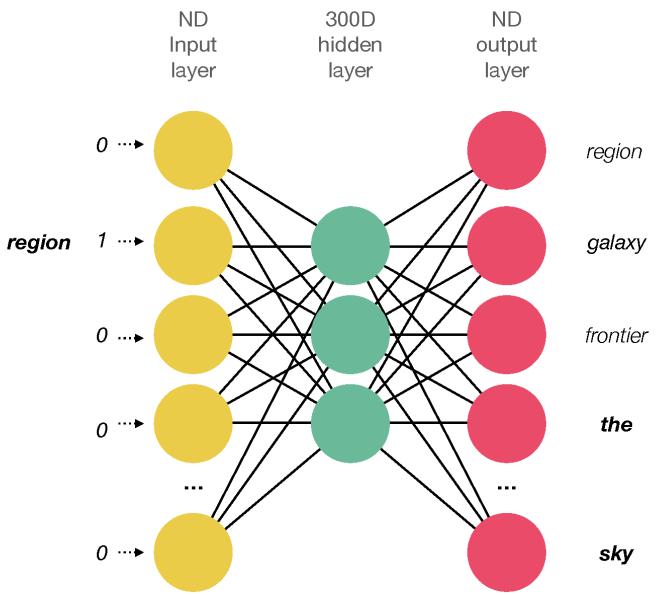
I called into the microphone:

"Station Solaris! Station Solaris! Station Solaris! I think I am leaving the flight-path, correct my course! Station Solaris, this is the _Prometheus_ capsule. Over."

I had missed the precious moment when the planet first came into view. Now it was spread out before my eyes; flat, and already immense. Nevertheless, from the appearance of its surface, I judged that I was still at a great height above it, since I had passed that imperceptible frontier after which we measure the distance that separates us from a celestial body in terms of altitude. I was falling. Now I had the sensation of falling, even with my eyes closed. (I quickly reopened them: I did not want to miss anything there was to be seen.)

(https://www.webnovel.com/book/solaris-2.0_25879657706474305)

Example



"You're on your way, Kelvin. Good luck!" Moddard's voice sounded as close as before.

A wide slit opened at eye-level, and I could see the stars. The _Prometheus_ was orbiting in the region of Alpha in Aquarius and I tried in vain to orient myself; a glittering dust filled my porthole. I could not recognize a single constellation; in this region of the galaxy the sky was unfamiliar to me. I waited for the moment when I would pass near the first distinct star, but I was unable to isolate any one of them. Their brightness was fading; they receded, merging into a vague, purplish glimmer, the sole indication of the distance I had already travelled. My body rigid, sealed in its pneumatic envelope, I was knifing through space with the impression of standing still in the void, my only distraction the steadily mounting heat.

Suddenly, there was a shrill, grating sound, like a steel blade being drawn across a sheet of wet glass. This was it, the descent. If I had not seen the figures racing across the dial, I would not have noticed the change in direction. The stars having vanished long since, my gaze was swallowed up on the pale reddish glow of infinity. I could hear my heart thudding heavily. I could feel the coolness from the air-conditioning on my neck, although my face seemed to be on fire. I regretted not having caught a glimpse of the _Prometheus_, but the ship must have been out of sight by the time the automatic controls had raised the shutter of my porthole.

The capsule was shaken by a sudden jolt, then another. The whole vehicle began to vibrate. Filtered through the insulating layers of the outer skins, penetrating my pneumatic cocoon, the vibration reached me, and ran through my entire body. The image of the dial shivered and multiplied, and its phosphorescence spread out in all directions. I felt no fear. I had not undertaken this long voyage only to overshoot my target!

I called into the microphone:

"Station Solaris! Station Solaris! Station Solaris! I think I am leaving the flight-path, correct my course! Station Solaris, this is the _Prometheus_ capsule. Over."

I had missed the precious moment when the planet first came into view. Now it was spread out before my eyes; flat, and already immense. Nevertheless, from the appearance of its surface, I judged that I was still at a great height above it, since I had passed that imperceptible frontier after which we measure the distance that separates us from a celestial body in terms of altitude. I was falling. Now I had the sensation of falling, even with my eyes closed. (I quickly reopened them: I did not want to miss anything there was to be seen.)

(https://www.webnovel.com/book/solaris-2.0_25879657706474305)

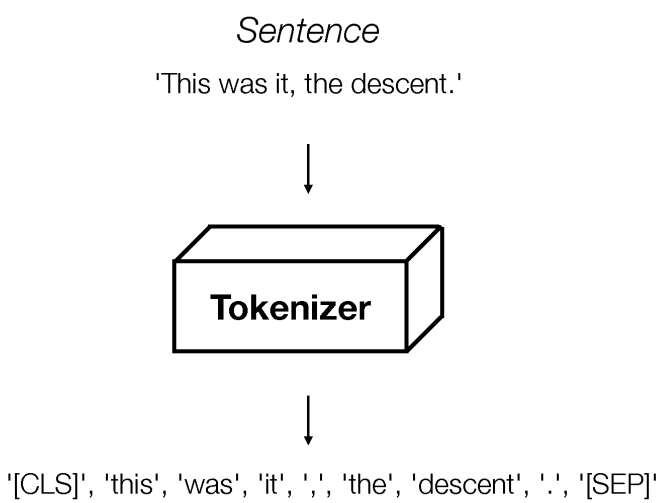
Transformers

Transformers

- Word2Vec: Generates **static** word embeddings
  $\Rightarrow$ The vector representation for a word is the same regardless of its context
  - Transformers: Generate **dynamic** word embeddings
    $\Rightarrow$ The vector representation for a word can change based on its context
- Word2Vec: Learns representations based on the **local** context (i.e., surrounding words)
  - Transformers: Learn representations based on the **entire** context of the sentence or even multiple sentences
- Most contemporary Large Language Models (LLMs) implement a transformer architecture, such as:
  - OpenAI's ChatGPT
  - Google's Gemini
  - Meta's LLaMA
  - Anthropic's Claude

Transformers

- Transformers rely on more advanced tokenizations:



"You're on your way, Kelvin. Good luck!" Moddard's voice sounded as close as before.

A wide slit opened at eye-level, and I could see the stars. The _Prometheus_ was orbiting in the region of Alpha in Aquarius and I tried in vain to orient myself; a glittering dust filled my porthole. I could not recognize a single constellation; in this region of the galaxy the sky was unfamiliar to me. I waited for the moment when I would pass near the first distinct star, but I was unable to isolate any one of them. Their brightness was fading; they receded, merging into a vague, purplish glimmer, the sole indication of the distance I had already travelled. My body rigid, sealed in its pneumatic envelope, I was knifing through space with the impression of standing still in the void, my only distraction the steadily mounting heat.

Suddenly, there was a shrill, grating sound, like a steel blade being drawn across a sheet of wet glass. This was it, the descent. If I had not seen the figures racing across the dial, I would not have noticed the change in direction. The stars having vanished long since, my gaze was swallowed up on the pale reddish glow of infinity. I could hear my heart thudding heavily. I could feel the coolness from the air-conditioning on my neck, although my face seemed to be on fire. I regretted not having caught a glimpse of the _Prometheus_, but the ship must have been out of sight by the time the automatic controls had raised the shutter of my porthole.

The capsule was shaken by a sudden jolt, then another. The whole vehicle began to vibrate. Filtered through the insulating layers of the outer skins, penetrating my pneumatic cocoon, the vibration reached me, and ran through my entire body. The image of the dial shivered and multiplied, and its phosphorescence spread out in all directions. I felt no fear. I had not undertaken this long voyage only to overshoot my target!

I called into the microphone:

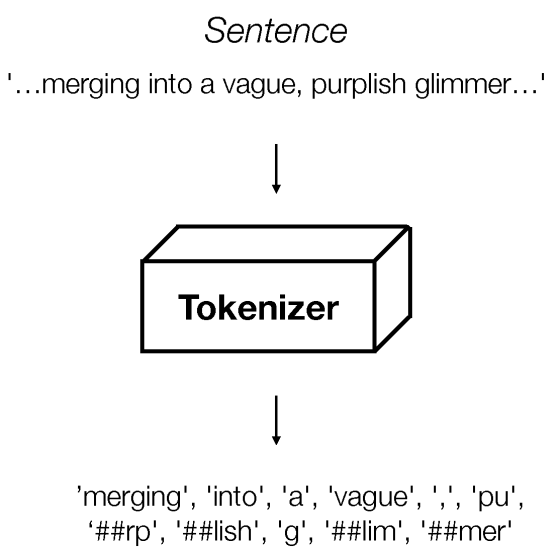
"Station Solaris! Station Solaris! Station Solaris! I think I am leaving the flight-path, correct my course! Station Solaris, this is the _Prometheus_ capsule. Over."

I had missed the precious moment when the planet first came into view. Now it was spread out before my eyes; flat, and already immense. Nevertheless, from the appearance of its surface, I judged that I was still at a great height above it, since I had passed that imperceptible frontier after which we measure the distance that separates us from a celestial body in terms of altitude. I was falling. Now I had the sensation of falling, even with my eyes closed. (I quickly reopened them: I did not want to miss anything there was to be seen.)

(https://www.webnovel.com/book/solaris-2.0_25879657706474305)

Transformers

- Transformers rely on more advanced tokenizations:



"You're on your way, Kelvin. Good luck!" Moddard's voice sounded as close as before.

A wide slit opened at eye-level, and I could see the stars. The _Prometheus_ was orbiting in the region of Alpha in Aquarius and I tried in vain to orient myself; a glittering dust filled my porthole. I could not recognize a single constellation; in this region of the galaxy the sky was unfamiliar to me. I waited for the moment when I would pass near the first distinct star, but I was unable to isolate any one of them. Their brightness was fading; they receded, merging into a vague, purplish glimmer, the sole indication of the distance I had already travelled. My body rigid, sealed in its pneumatic envelope, I was knifing through space with the impression of standing still in the void, my only distraction the steadily mounting heat.

Suddenly, there was a shrill, grating sound, like a steel blade being drawn across a sheet of wet glass. This was it, the descent. If I had not seen the figures racing across the dial, I would not have noticed the change in direction. The stars having vanished long since, my gaze was swallowed up on the pale reddish glow of infinity. I could hear my heart thudding heavily. I could feel the coolness from the air-conditioning on my neck, although my face seemed to be on fire. I regretted not having caught a glimpse of the _Prometheus_, but the ship must have been out of sight by the time the automatic controls had raised the shutter of my porthole.

The capsule was shaken by a sudden jolt, then another. The whole vehicle began to vibrate. Filtered through the insulating layers of the outer skins, penetrating my pneumatic cocoon, the vibration reached me, and ran through my entire body. The image of the dial shivered and multiplied, and its phosphorescence spread out in all directions. I felt no fear. I had not undertaken this long voyage only to overshoot my target!

I called into the microphone:

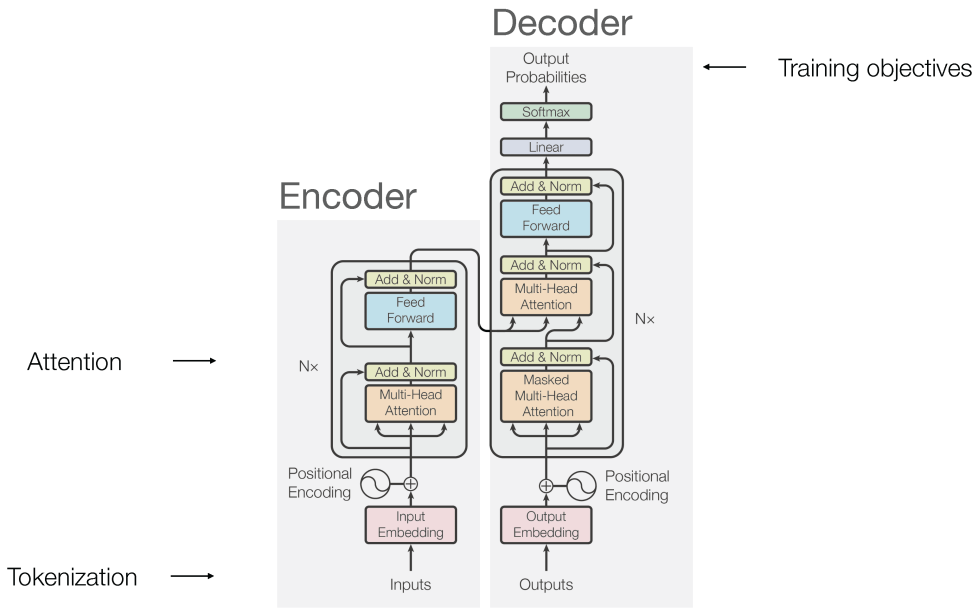
"Station Solaris! Station Solaris! Station Solaris! I think I am leaving the flight-path, correct my course! Station Solaris, this is the _Prometheus_ capsule. Over."

I had missed the precious moment when the planet first came into view. Now it was spread out before my eyes; flat, and already immense. Nevertheless, from the appearance of its surface, I judged that I was still at a great height above it, since I had passed that imperceptible frontier after which we measure the distance that separates us from a celestial body in terms of altitude. I was falling. Now I had the sensation of falling, even with my eyes closed. (I quickly reopened them: I did not want to miss anything there was to be seen.)

(https://www.webnovel.com/book/solaris-2.0_25879657706474305)

Excurse: Components of Transformers

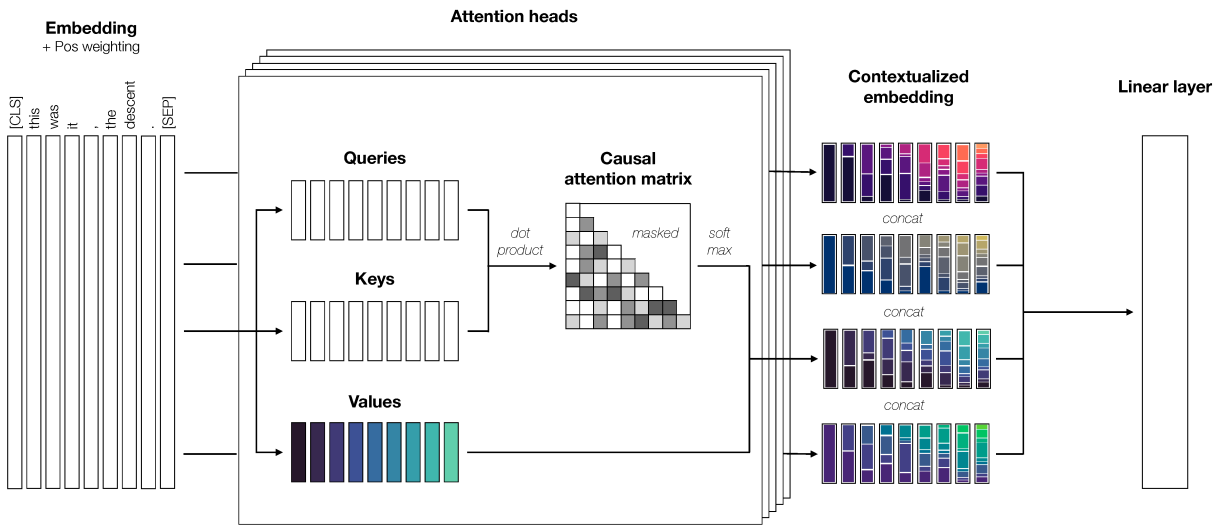
- At each layer, each token is contextualized within the scope of the context window with other (unmasked) tokens
  - This is done via a parallel multi-head attention mechanism, which allows the signal for key tokens to be amplified and the one for less important tokens to be diminished



(Vaswani et al., 2017)

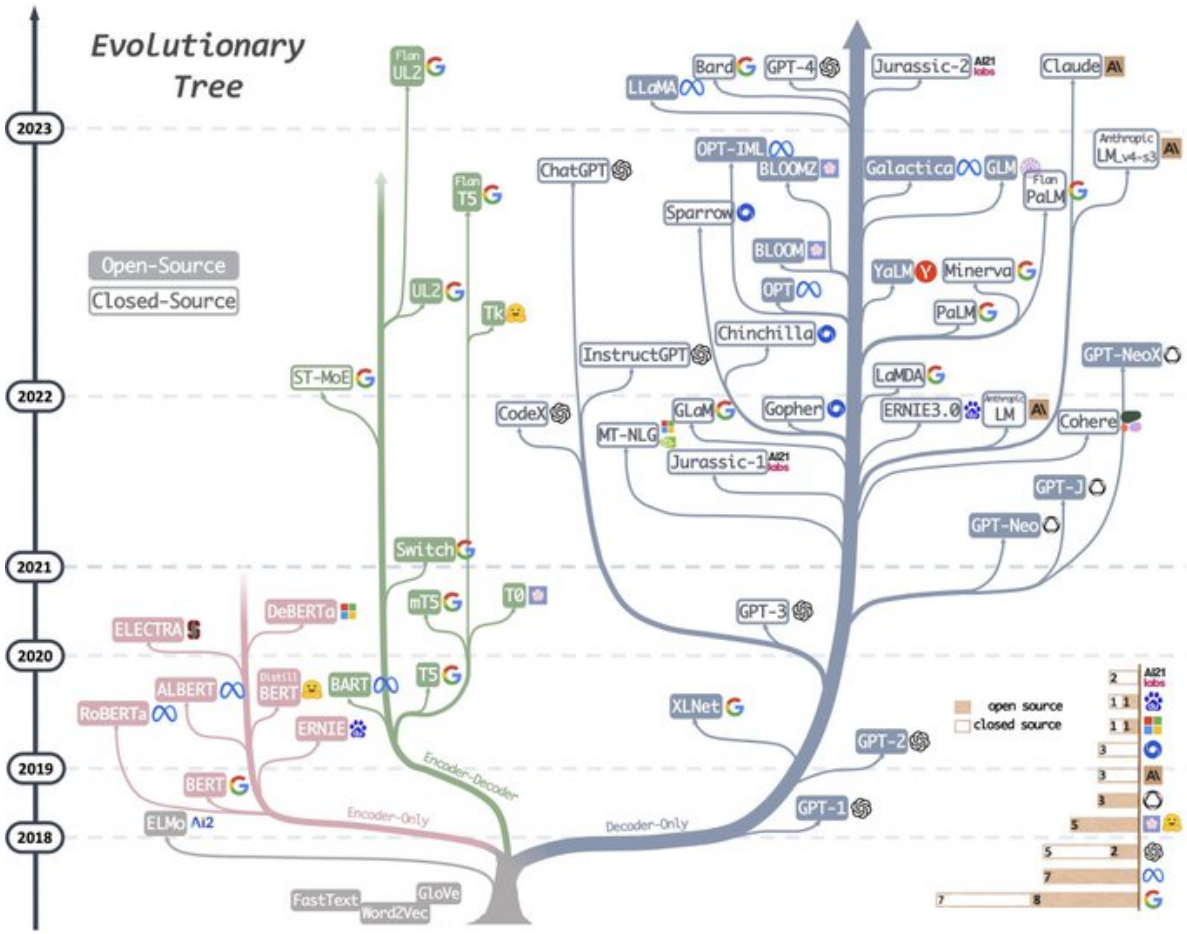
Excurse: Attention Mechanism of Transformers

- Multi-head attention: Splitting the input embeddings into multiple "heads," each learning different types of information, and calculating attention scores that are used to weight the embedding vectors
  - The outputs from all heads are concatenated and transformed to produce the final output



(adapted from https://x.com/dirkuwulff/status/1694988985225839048)

Excurse: Evolution of Transformers



(https://x.com/DrJimFan/status/1651968203701231616)

Transformers from Python in R

- Sentiment classification task:
  - We will use DistilBERT, which is a light and fast transformer LLM trained by distilling Google's BERT (Bidirectional Encoder Representations from Transformers), which is the predecessor of Gemini

```
library(reticulate) #; py_config() #; use_condaenv("r-reticulate")

# Loading the tokenizer (i.e., pre-processor) and transformer model:
transformers <- import("transformers") #import Python package as "variable" into R
tkn <- transformers$AutoTokenizer$from_pretrained('distilbert-base-uncased-finetuned-sst-2-
ppl <- transformers$pipeline(model = 'distilbert-base-uncased-finetuned-sst-2-english'
                             , tokenizer = tkn)
ppl
## <transformers.pipelines.text_classification.TextClassificationPipeline object at 0x00000
##  signature: (inputs, **kwargs)

# Exemplary sentiment predictions:
ppl(c("I like you!", "I don't like you!")) %>% as.data.frame()
##      label     score  label.1   score.1
## 1 POSITIVE 0.9998792 NEGATIVE 0.9989811
```

Transformers from Python in R

- The `emotion` dataset from Hugging Face contains English tweets that are classified according to six basic emotions: joy, love, anger, fear, sadness, and surprise

```
datasets <- import("datasets") #import Python package as "variable" into R
dat = datasets$load_dataset("dair-ai/emotion", split = 'test')

# Translate true numeric class to verbal label and transforming Python to R data:
dict_emo <- dat$features$label$names
dat <- dat$to_pandas() %>%
  as_tibble() %>%
  mutate(emotion = dict_emo[label+1]) #NOTE: indexing starts at 0 in Python (vs. 1 in R)

# Selecting a random subset of tweets:
set.seed(1)
dat_subset <- dat %>% sample_n(100)
tail(dat_subset)
## # A tibble: 6 x 3
##   text                                                      label emotion
##   <chr>                                                     <dbl> <chr>
## 1 im sorry im feeling a little bitchy tacky looking women came in~     3 anger
## 2 i hope everyone can help with charity work without feeling stre~     0 sadness
## 3 i feel lost without you                                      0 sadness
## 4 i definitely feel there s some useful information here for anyo~     1 joy
## 5 i can imagine most young people might feel resentful about the ~     3 anger
```

Transformers from Python in R

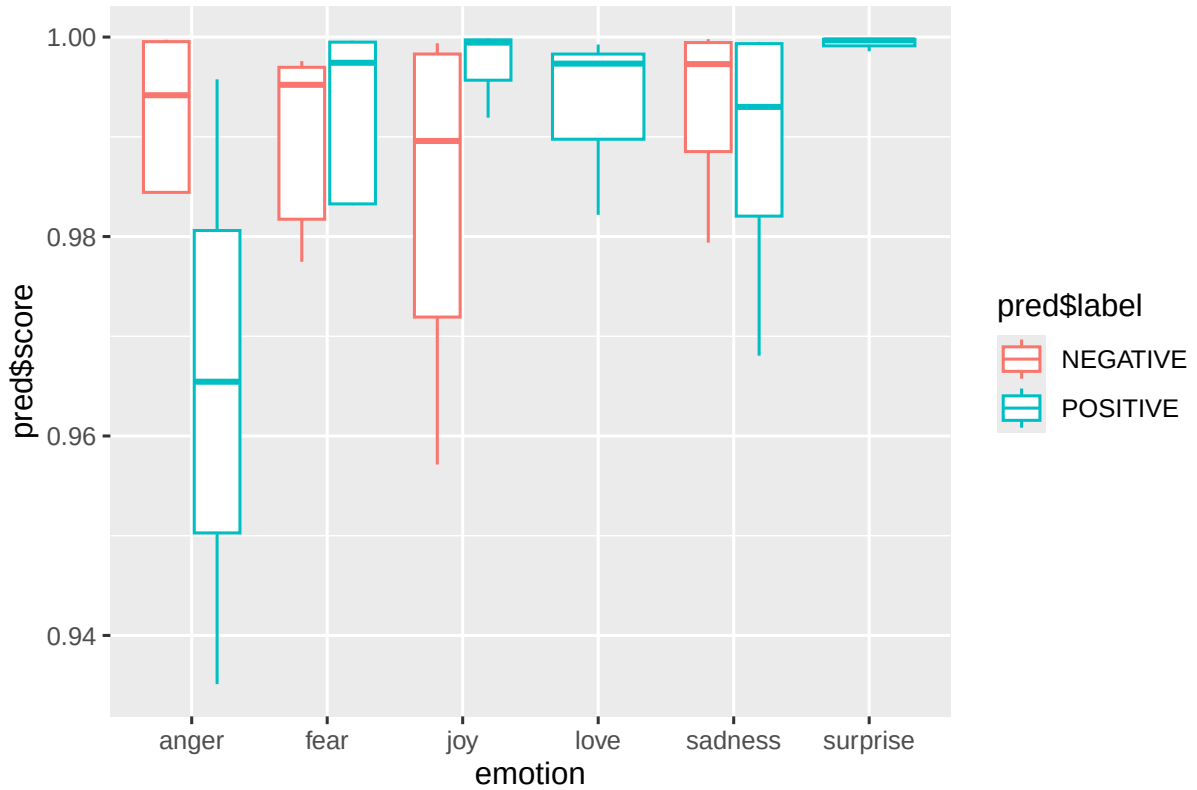
- We can use DestilBERT to classify the tweets as positive or negative sentiment:

```
# Predictions for all tweets:
dat_subset <- dat_subset %>%
  rowwise() %>%
  mutate(pred = ppl(text) %>% as.data.frame())
tail(dat_subset)
## # A tibble: 6 x 4
## # Rowwise:
##   text                                                label emotion pred$label
##   <chr>                                               <dbl> <chr>   <chr>
## 1 im sorry im feeling a little bitchy tacky looking wo~     3 anger   NEGATIVE
## 2 i hope everyone can help with charity work without f~     0 sadness POSITIVE
## 3 i feel lost without you                                   0 sadness NEGATIVE
## 4 i definitely feel there s some useful information he~     1 joy     NEGATIVE
## 5 i can imagine most young people might feel resentful~     3 anger   NEGATIVE
## 6 i am a prolific writer in my fandom but do not feel ~     1 joy     POSITIVE
## # i 1 more variable: pred$score <dbl>
```

Transformers from Python in R

- Comparison of predicted sentiment, separately for each of the six actual basic emotions:

```
ggplot(dat_subset, aes(x = emotion, y = pred$score, color = pred$label)) +
  geom_boxplot(outliers = FALSE)
```



Excurse: Generative AI from Python in R

- Google's BERT:

```
# Loading the tokenizer and transformer model:
tkn_bert <- transformers$AutoTokenizer$from_pretrained('distilbert/distilgpt2')
ppl_bert <- transformers$pipeline(model = 'distilbert/distilgpt2',
                                  tokenizer = tkn_bert)
ppl_bert
## <transformers.pipelines.text_generation.TextGenerationPipeline object at 0x0000021C811A5
##  signature: (text_inputs, **kwargs)

# Exemplary text generation:
pred_bert <- ppl_bert("My name is Tobias and I am teaching a class on Machine Learning")
pred_bert[[1]]$generated_text %>% strwrap(., width = 80)
## [1] "My name is Tobias and I am teaching a class on Machine Learning. For instance,"
## [2] "my last question will be: \"How do I have access to your data?\" I answer that"
## [3] "question. The answer is always that I'll be able to use these"
```

Excurse: Generative AI from Python in R

- OpenAI's GPT:

```
# Loading the tokenizer and transformer model:
tkn_gpt2 <- transformers$AutoTokenizer$from_pretrained('openai-community/gpt2-medium')
ppl_gpt2 <- transformers$pipeline(model = 'openai-community/gpt2-medium',
                                  tokenizer = tkn_gpt2)
ppl_gpt2
## <transformers.pipelines.text_generation.TextGenerationPipeline object at 0x0000021C83C49
##  signature: (text_inputs, **kwargs)

# Exemplary text generation:
pred_gpt2 <- ppl_gpt2("My name is Tobias and I am teaching a class on Machine Learning")
pred_gpt2[[1]]$generated_text %>% strwrap(., width = 80)
## [1] "My name is Tobias and I am teaching a class on Machine Learning."
## [2] ""
## [3] "This class will cover the following topics:"
## [4] ""
## [5] "The core concepts are:"
## [6] ""
## [7] "Deep Learning, classification, classification accuracy, regression, predictive"
## [8] "models, clustering, and"
```

PA-Projects

Schedule

Date	Topics / Work Program	
	FS	FP
18.04.2024	Module 1	
25.04.2024	Module 1 + Module 2	
02.05.2024	Module 2	
09.05.2024	*Public Holiday*	
16.05.2024	Module 3	Group Formation
23.05.2024	*Spring Break*	
30.05.2024	*Public Holiday*	
06.06.2024	Module 4	Project Planning
13.06.2024	Module 5	Project Finalization
20.06.2024	Proposal Presentations	
27.06.2024		Project Work
04.07.2024		Project Work
11.07.2024		Final Presentations
18.07.2024	Summary & Wrap-Up	
25.07.2024	*Buffer*	
30.09.2024	Submission Deadline: Written Summary	

Suggested Structure of Proposal Presentations

1. Theoretical background and motivation of the original study
2. Brief summary of the (most important) original results
   - Ideally using original graphs/tables for quick and easy comprehension
3. Detailed description of the original research methodology
4. Detailed description of the planned reanalysis
   - Including feasibility assessment
5. Speculation on potential/desired outcomes
   - E.g., new substantive and methodological insights
6. Open questions

- Duration: **20-25 minutes**
- Format: Rmd, Powerpoint, ...

Group Project Planning

- Current state of your project ideas?
  - Any specific issues/questions?
- Does Aka and Bhatia (2022) clarify the expected format and scope of the written summaries?

Potential Project Suggestions

- Data from this year's Bachelor-ExPra project: Continued influence effect (after retraction of certain news articles) with 5 open text responses
  - Originally to be manually classified by ExPra-students as exhibiting continued influence despite later information that the article was wrong
  - Incl. sentiment analysis (e.g., use of specific keywords like "the victim was murdered **in cold blood**")
- Custom LLM (e.g., GPT) that actively neglects updated beliefs in multi-shot prompting interactions for, e.g., political events forecasting
- Machine Learning Weights of Advice (ML-WOA): ML-based extension of Mixed-Effects Regression Weights of Advice (MER-WOA; Rebholz et al., 2024) with multiple options:
  1. Shapley values
  2. Based on individual conditional expectation plots (see also Pargent et al., 2023)
  3. ???

Summary

Summary

- DL is attractive for:
  - Extremely large training sample sizes
  - Low priority of model interpretability ($\rightarrow$ next week!)
- NNs are particularly suited for niche tasks, such as:
  - Image recognition
  - Speech and text modeling
- Classic NLP tasks: E.g., sentiment analysis, but also things like language translation
  - Recurrent NNs and transformers have shown great success in these areas due to their ability to handle sequential data and capture long-term dependencies

Homework

- FS:
  - Finish/Revisit the programming tutorials
  - Readings for next week:
    - Taylor, J. E. T., & Taylor, G. W. (2021). Artificial cognition: How experimental psychology can help generate explainable artificial intelligence. *Psychonomic Bulletin & Review*, *28*(2), 454–475. https://doi.org/10.3758/s13423-020-01825-5
- FP:
  - Continuing the group project planning
    - **To be finalized next week!**
  - Starting to prepare the proposal presentation of your group project