Data mining and machine learning

09.07.2025

- Prof Francesco IARLORI

IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-Mission 4 “Education and Research” - Component 2: “From research to business” - Investment 3.1: “Fund for the realisation of an integrated system of research and innovation infrastructures”

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani

Day 2

Time	Duration	Training Module - Topic
09:00 - 09:30	30m	Welcome & Introduction to Environmental AI
09:30 - 10:30	1h	Module 1: Biodiversity Monitoring with AI
10:30 - 10:45	15m	Coffee Break
10:45 - 11:45	1h	Module 2: Climate Modelling and Extreme Events Prediction
11:45 - 13:00	1h15m	Module 3: Remote Sensing with AI
13:00 - 14:00	1h	Lunch Break
14:00 - 15:30	45m	Module 4: Big Environmental Data Analysis
15:30 – 16:00	30m	Module 5: Language Models & NLP in Research
15:30 - 15:45	15m	Coffee Break
15:45 - 16:30	45m	Final Activity: Case Studies & Group Work

Welcome & Introduction to Environmental AI

- Quick recap of Day 1 (AI/ML basics)
- Why AI matters for environmental challenges
- Overview of today's topics and tools

What is a Model in AI

- In Artificial Intelligence (AI), a model is a mathematical or computational representation of a system that can make predictions or decisions based on data.

Algorithm vs Model

- Algorithm: The procedure or recipe (e.g., decision tree, neural network).
- Model: The result of applying the algorithm to data (trained with weights, rules, or structure).

- Training: The process of feeding data into an algorithm so it can learn patterns.
- Inference: Once trained, the model is used to make predictions or decisions.

Example

- If you want an AI to recognize cats in images:
- You start with an algorithm (like a convolutional neural network).
- You train it on thousands of labeled images (some with cats, some without).
- After training, the output is a model that can identify cats in new pictures.

Types of Models in AI

- Linear Regression (for predictions)
- Decision Trees (for classification)
- Neural Networks (for complex tasks like image or speech recognition)
- Clustering Models (for grouping similar data)
- Language Models (like ChatGPT)

What Are Hugging Face Models ?

These are ready-to-use AI models hosted on the Hugging Face Model Hub. They are trained on large datasets and can be easily reused or fine-tuned for tasks like:

- Text classification (e.g., spam detection)
- Sentiment analysis
- Question answering
- Translation
- Text generation (e.g., using GPT, T5, BERT)
- Image classification and speech recognition (more recently)

Hugging Face

Tools Offered by Hugging Face

- Transformers – Library for using and training models.
- Datasets – Access to thousands of datasets.
- Hub – Online repository for models and datasets.
- Inference API – Deploy models with an API, no infrastructure setup.
- AutoTrain – No-code training interface.

- Hugging Face models make advanced AI capabilities (especially language understanding and generation) more accessible, reusable, and customizable. They play a central role in speeding up AI research, development, and deployment.

What Role Do They Play?

- **Accelerate AI Development**
  - You don’t have to train models from scratch - Hugging Face offers thousands of pretrained models.
- **Democratize AI**
  - It makes cutting-edge models accessible to researchers, developers, and businesses through an open platform.
- **Standardize Interfaces**
  - Using transformers, a Hugging Face library, you can load and use models with just a few lines of code
- **Support Fine-Tuning**
  - You can easily customize models for your specific domain (legal, medical, etc.) using transfer learning.
- **Enable Collaboration**
  - The Hugging Face hub works like GitHub — users can upload, version, and share their models.
- **Power Inference at Scale**
  - With transformers, datasets, and accelerate, Hugging Face supports training and deploying models on CPUs, GPUs, and TPUs, even in production.

AI Alghorithms

Module 1: Biodiversity Monitoring with A

- Species recognition from audio (e.g., birdsong) and images (e.g., camera traps)
- Case studies (e.g., Cornell Lab of Ornithology, iNaturalist)
- Tools: Pretrained models, Google Teachable Machine, custom image classifiers
- 📌 Activity: Group discussion – What biodiversity challenges could benefit from AI?

IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-Mission 4 "Education and Research" - Component 2: "From research to business" - Investment 3.1: "Fund for the realisation of an integrated system of research and innovation infrastructures"

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani

Introduction & Motivation

- Why recognizing species matters (conservation, biodiversity monitoring, citizen science)
- Challenges: manual data collection, human limitations

Species Recognition from Audio

- What is bioacoustics?
- How birdsong and animal sounds are used for identification
- Key features: pitch, rhythm, spectrograms

Species Recognition from Images

- Role of camera traps and smartphone photos
- Features: colour, shape, patterns
- Detection vs classification

Case Study – Cornell Lab of Ornithology

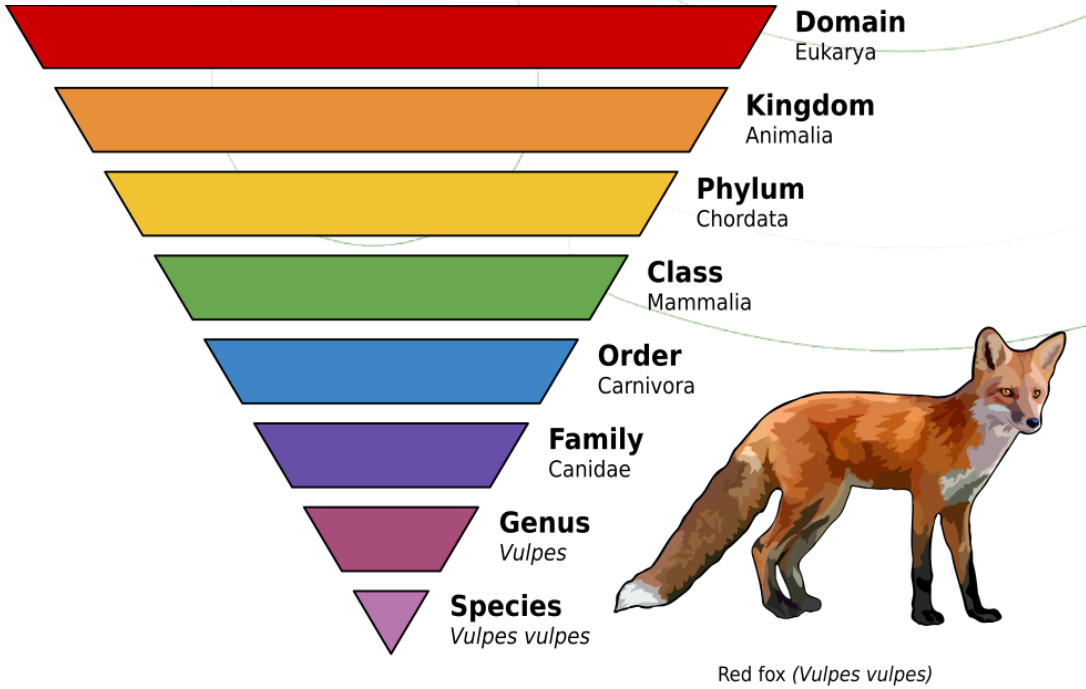
- Merlin Bird ID: how it identifies birds from sound
- Macaulay Library: large-scale annotated datasets
- Machine learning behind the scenes

Case Study – iNaturalist

- Citizen science at scale
- How users contribute and models learn from labelled data
- Collaboration with researchers

Tools – Pretrained Models

- Introduction to pretrained models (e.g., ResNet, YAMNet)
- Advantages: faster development, lower data needs
- Examples of species classification models



Red fox *(Vulpes vulpes)*

Tools – Google Teachable Machine

- Overview of Teachable Machine
- Quick demo idea: train a bird vs. cat classifier
- Pros & cons (accessibility vs flexibility)

Tools – Custom Image and Audio Classifiers

- Using TensorFlow or PyTorch for fine-tuning
- Steps: data collection, preprocessing, training, deployment
- When to build custom vs use existing tools

Real-World Applications

- Wildlife monitoring
- Invasive species detection
- Habitat change analysis

Challenges & Limitations

- Data quality and bias
- Generalization across regions and seasons
- Need for human validation

Future Directions

- Edge AI for in-field recognition
- Multimodal models (audio + image)
- Citizen science platforms powered by AI

Recap & Key Takeaways

- Summary of tools, techniques, and real-world value
- Empowering both researchers and the public

- See Ex01

Feature	ResNet50 (ImageNet)	iNaturalist Model (e.g., ResNet50)
Classes	1,000 general categories	5,000+ species-level labels
Training Data	ImageNet	iNaturalist (real species photos)
Output Layer	Dense(1000)	Dense(N_species)
Purpose	General object classification	Biodiversity, ecological image classification

Resources & Links

Recommended reading

- Practical Deep Learning for Coders – Jeremy Howard & Sylvain Gugger
  Great for hands-on learning with image classification and transfer learning.
- Deep Learning with Python" – François Chollet
  Covers CNNs and sound/image classification with Keras.

Research Articles & Case Studies:

- "Bird species recognition using machine learning with spectrograms"
  (Check journals like Ecological Informatics, PLOS ONE, or Bioacoustics)
- Papers with Code - Sound Classification Benchmarks:
  👉 https://paperswithcode.com/task/audio-classification

Open Datasets

- Cornell's Macaulay Library Dataset
  - Includes bird sound and image datasets. Often used with BirdNET. https://www.macaulaylibrary.org
- Xeno-Canto – Community-collected bird sound dataset
  - https://www.xeno-canto.org
- iNaturalist Datasets
  - Official datasets from iNaturalist competitions (e.g., FGVC) https://www.kaggle.com/c/inaturalist-2019-fgvc6/data
- Google AudioSet
  - Large-scale dataset of sound events including animal sounds https://research.google.com/audioset/
- DCASE Challenges – Acoustic scene and sound event datasets
  - http://dcase.community/challenge2023

Module 2: Climate Modelling and Extreme Events Prediction

- Using AI/ML for climate simulations, trend detection
- Predicting floods, wildfires, heatwaves
- Datasets and challenges (e.g., low resolution, uncertainty)
- Use of deep learning models (CNNs, RNNs) in time-series forecasting
- 📌 Mini-case study: AI predicting cyclone intensity or droughts

IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-Mission 4 “Education and Research” - Component 2: “From research to business” - Investment 3.1: “Fund for the realisation of an integrated system of research and innovation infrastructures”

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani

Introduction & Importance - Ex02

- Climate change: urgency of better forecasting
- Traditional climate models: strengths and gaps
- Where AI/ML fits in

From Simulations to Predictions

- Definitions:
  - Climate simulation (long-term)
  - Weather prediction (short-term)
- AI/ML as complementary tools to physics-based models

Detecting Trends & Anomalies

- Long-term climate trend detection using ML
- AI in analyzing temperature, sea level, CO_2 data
- Time-series insights beyond human capability

Predicting Extreme Events

- AI for early warning: floods, wildfires, heatwaves
- Need for high spatial/temporal precision
- Role of supervised learning, ensemble methods

Mini-Case Study – Cyclone Intensity & Drought Prediction

- Use of deep learning (e.g., RNNs, LSTMs) to model cyclone paths & intensities
- Satellite data + historical events
- Drought classification using CNNs with soil moisture and precipitation maps

Climate Datasets & Their Challenges

- Common datasets: ERA5, MODIS, CMIP6, NASA POWER
- Issues:
  - Low spatial resolution
  - Incomplete records
  - Noise & uncertainty
- Need for data harmonization

Climate Datasets

- CHELSA (Climatologies at High Resolution for the Earth's Land Surface Areas)
  - What it is: High-resolution (30 arc-seconds) climate data based on statistical downscaling of reanalysis data.
  - Includes: Historical and future climate scenarios (CMIP5/CMIP6).
  - Great for: Mountainous regions and biodiversity applications.
- NASA POWER
  - What it is: Daily weather and solar data from satellite and modeled sources.
  - Tools: Download via API or web GUI.
  - Variables: Temperature, humidity, wind speed, solar radiation
- Copernicus Climate Data Store (CDS)
  - What it is: Comprehensive data from ECMWF (European Centre for Medium-Range Weather Forecasts).
  - Includes: ERA5 reanalysis, seasonal forecasts, and more.
  - ERA5: Global climate reanalysis at ~30 km resolution, hourly data available.

Deep Learning for Time-Series Forecasting

- Recurrent Neural Networks (RNNs), LSTMs for sequential climate signals
- CNNs for gridded atmospheric/satellite imagery
- Example: Rainfall prediction using ConvLSTM

AI + Physics = Hybrid Climate Models

- Physics-informed neural networks (PINNs)
- Learning residuals of physics-based models
- Better generalization & interpretability

Tools & Platforms

- Tools: TensorFlow, PyTorch, xarray, ClimateLearn
- Google Earth Engine, Copernicus Climate Data Store
- Access to HPC + cloud resources for training models

Open Projects & Benchmarks

- ClimateBench: ML benchmark dataset for climate variables
- AI for Earth, DeepMind's GraphCast
- ECMWF + NVIDIA AI models

Challenges & Ethical Considerations

- Data bias, overfitting in ML models
- Black-box risks in high-stakes decisions
- Fair access to AI tools for global south

Future Directions

- Multimodal learning (e.g., combining weather maps + text + radar)
- Transfer learning across regions
- Citizen AI for community-based forecasting

Key Takeaways

- AI accelerates insights from complex climate data
- Not a replacement, but a powerful ally to traditional models
- Open collaboration is key to progress

Q&A / Discussion

- Invite questions
  - Prompt: “What’s a climate application in your region that could benefit from AI?”

Module 3: Remote Sensing with AI

- Satellite, drone, and LIDAR data sources
- Automated image classification and land use analysis
- Tools: Google Earth Engine + Python; QGIS with ML plugins
- 📌 Demo: Land cover classification using Earth Engine + ML
- 📌 Group work: Brainstorm ideas for AI-powered remote sensing applications

IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-Mission 4 “Education and Research” - Component 2: “From research to business” - Investment 3.1: “Fund for the realisation of an integrated system of research and innovation infrastructures”

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani

Why Remote Sensing + AI?

- Importance of environmental monitoring and land use mapping
- Limitations of manual analysis at large scales
- AI’s role in accelerating insights from remote sensing data

Data Sources Overview

- Satellites: Sentinel, Landsat, MODIS
- Drones: High-res, local observations
- LIDAR: 3D terrain and vegetation structure
- Trade-offs: resolution, coverage, update frequency

What AI Can Do with This Data

- Automated land cover classification
- Change detection (deforestation, urban expansion)
- Object detection (buildings, roads, crops)
- Terrain analysis using elevation and canopy structure

Google Earth Engine (GEE)

- Cloud-based geospatial processing platform
- Integrates global satellite archives
- Machine Learning integration (random forest, CART, SVM, TensorFlow)
- Visual programming and Python API

Python & QGIS Tools

- Python: scikit-learn, TensorFlow, geemap for Earth Engine
- QGIS: Semi-Automatic Classification Plugin, ML plugins
- Combining open-source tools for end-to-end workflows

Demo – Land Cover Classification with Earth Engine

- Step-by-step:
  - Load Sentinel-2 imagery
  - Train classifier (e.g., Random Forest)
  - Visualize classified map (forest, water, urban, agri)
- Option: Compare with ground truth or historical images

Key Considerations in Classification

- Selecting good training samples
- Handling cloud cover and seasonal changes
- Accuracy assessment: confusion matrix, kappa coefficient
- Edge cases (e.g., mixed land types)

Use Cases & Impact

- Forest monitoring (REDD+)
- Urban planning and smart cities
- Disaster response (flood zones, landslides)
- Agriculture (crop health, yield estimation)

Group Activity - AI + Remote Sensing Idea Storm

- Prompt: “Design a remote sensing project using AI for your local environment. What problem would it solve?”
- Groups of 2–3
- 10 min brainstorm
- 2 min per group shareback

Challenges & Ethics

- Data access inequality
- Surveillance and privacy risks (especially with drones)
- Interpretability and trust in ML decisions
- Need for local validation

Resources & Datasets

- Earth Engine datasets: Sentinel, Landsat, CHIRPS, etc.
- LIDAR: USGS 3DEP, OpenTopography
- Drones: OpenDroneMap, DroneDeploy
- Tutorials: Google Dev Docs, Mapbox, Radiant Earth

Recap & Takeaways

- AI + remote sensing unlocks global and local-scale insights
- Earth Engine + Python/QGIS = powerful, accessible combo
- Always balance automation with ground truth and ethics

What's Next?

- Learn GEE Python API
- Try QGIS with ML
- Explore open competitions: DrivenData, Zindi, SpaceNet

Q&A / Feedback

- Invite discussion
- Ask: “What application would you prototype if you had a satellite feed for your neighborhood?”

Module 4: Big Environmental Data Analysis

- Handling air/water/soil quality datasets
- Data cleaning, feature extraction, anomaly detection
- Libraries: Pandas, Scikit-learn, TensorFlow for environmental data
- 📌 Example: Detecting pollution anomalies using ML in air quality data

IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-Mission 4 “Education and Research” - Component 2: “From research to business” - Investment 3.1: “Fund for the realisation of an integrated system of research and innovation infrastructures”

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani

Why Analyse Environmental Quality Data?

- Environmental health and public safety depend on accurate, timely monitoring
- Massive sensor datasets are hard to interpret manually
- ML helps identify patterns, outliers, and trends at scale

Environmental Data Sources

- Air: PM2.5, NO_2, O_3 sensors (e.g., AirNow, OpenAQ)
- Water: pH, turbidity, contaminants (e.g., USGS, EEA)
- Soil: nutrient levels, moisture, toxins (e.g., FAO, SoilGrids)
- Time-series, geospatial, multivariate

Workflow Overview

- Ingest data (CSV, APIs, sensor streams)
- Clean & preprocess (handle missing data, normalize)
- Extract features (trends, rolling averages, domain transformations)
- Train ML models (classification, regression, clustering)
- Detect anomalies & generate alerts

Data Cleaning Essentials

- Handling missing values (interpolation, dropping, imputation)
- Dealing with noise and outliers
- Standardizing timestamps and units
- Tools: pandas, numpy, scikit-learn.preprocessing

Feature Engineering & Domain Context

- Rolling means, gradients, time-of-day effects
- Derived variables: AQI from raw pollutants
- Encoding seasonality, weather influence
- Use domain knowledge to engineer meaningful features

Example – Detecting Anomalies in Air Quality

- Dataset: PM2.5 and NO_2 sensor data from OpenAQ
- Model: Isolation Forest or One-Class SVM
- Steps:
  - Normalize and window the data
  - Train unsupervised model
  - Flag abnormal spikes not explained by season/weather
- Visual output: line chart with anomaly markers

Typical Air Quality Guidelines (WHO)

Feature	PM2.5	PM10	NO_2
What it is	Fine particles	Coarse particles	Gas (Nitrogen dioxide)
Size	≤ 2.5 µm	≤ 10 µm	Molecule (not a particle)
Source	Combustion	Dust, pollen	Combustion (vehicles, industry)
Health risk	High	Medium	High (especially respiratory)

See Ex03

ML Models for Environmental Data

- Supervised: regression (e.g., predict pollutant levels), classification (safe vs hazardous)
- Unsupervised: clustering, anomaly detection
- Time-series: LSTM for forecasting pollutant trends
- Model selection based on task + data availability

Python Tools & Libraries

- pandas: data wrangling
- scikit-learn: models, pipelines, validation
- tensorflow/keras: deep learning (LSTM, autoencoders)
- matplotlib, seaborn, plotly: visualizations
- prophet, tsfresh: time-series specific tools

Use Cases in Action

- Predicting PM2.5 spikes for health advisories
- Detecting illegal discharges in rivers
- Identifying agricultural soil depletion
- AI-powered environmental monitoring dashboards

Challenges in Environmental ML

- Noisy sensors and missing values
- External confounders (e.g., traffic, wind)
- Label scarcity for supervised models
- Data drift: models degrade over time without retraining

Best Practices & Validation

- Cross-validation for time-series (e.g., TimeSeriesSplit)
- Correlate anomalies with known events or news reports
- Alert thresholds + expert feedback loops
- Emphasize explainability and transparency

Resources & Datasets

- Datasets: OpenAQ, EPA, Water Quality Portal, SoilGrids
- Tutorials: Earth Data Science, DataCamp (environmental ML), Google Colab notebooks
- Community: GitHub, Kaggle, DrivenData environmental challenges

Takeaways

- ML unlocks hidden insights in environmental quality data
- Start small: clean data → extract features → anomaly detection
- Python ecosystem offers robust tools for all stages

Q&A and Discussion Prompt

- Prompt:
  - “What environmental issue in your community could benefit from ML-based monitoring?”
- Invite questions, feedback, shared concerns

Module 5: Language Models & NLP in Research

- Using Large Language Models (LLMs) to summarize scientific literature
- Tools: ChatGPT, SciSummary, PaperQA
- Applications: literature reviews, data extraction from reports
- 📌 Activity: Try summarizing a scientific paper using an LLM

IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-Mission 4 “Education and Research” - Component 2: “From research to business” - Investment 3.1: “Fund for the realisation of an integrated system of research and innovation infrastructures”

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani

Why Summarize Scientific Papers with AI?

- Explosion of publications: thousands of new papers daily
- Manual reading ≠ scalable for literature reviews
- LLMs help extract, condense, and synthesize key findings quickly

What Are Large Language Models (LLMs)?

- Definition: Trained on massive text corpora to understand and generate natural language
- Examples: ChatGPT, Claude, Gemini
- Strengths: abstraction, summarization, question answering
- Limitations: hallucination, outdated knowledge

Use Cases in Scientific Research

- Rapid literature reviews
- Extracting methods/results from PDFs
- Building databases from unstructured text
- Assisting grant writing or project scoping

Tools for Summarizing Papers

- ChatGPT: General-purpose, conversational
- SciSummary: Tailored for academic abstracts & summaries
- PaperQA: Ask targeted questions from a paper (retrieval-augmented)
- Bonus: Semantic Scholar, ScholarAI plugins, Scite Assistant

Comparison – LLM Summarization Styles

Tool	Summary Type	Strengths	Notes
ChatGPT	Conversational summary	Versatile, dialog-driven	Customizable prompts
SciSummary	Formal abstract-style	Fast, academic tone	Limited interactivity
PaperQA	Q&A from PDF chunks	Deep focus, source-linked	Needs good questions

Choose a short academic article or abstract (1–2 pages)

- Use ChatGPT or SciSummary to:
  - Generate a plain-language summary
  - Extract the main research question, method, and conclusion
- Discuss: Did the summary match your interpretation?

Prompt Engineering Tips

- Be specific: “Summarize the methodology section in bullet points.”
- Use roles: “You are a science journalist. Summarize this paper for a lay audience.”
- Ask for structured output: TL;DR, pros/cons, RQ + results

Automating Literature Reviews

- Build pipelines:
  - Search → scrape → summarize → tag
  - Zotero + API tools + LLM = semi-automated workflows
- Organize summaries into structured formats (e.g., CSV, Notion, Roam)

Limitations and Ethical Use

- LLMs may miss nuance or misinterpret data
- Don’t rely on summaries for critical decisions without reading the original
- Citing AI output? Follow institutional and journal policies

Integration with Other Research Tools

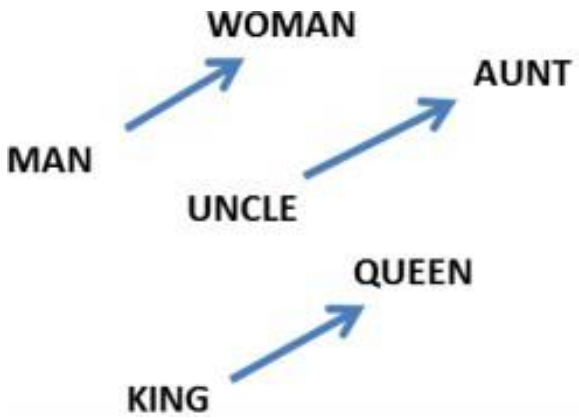
- Reference managers: Zotero, EndNote, Mendeley
- Markdown + GPT for research journaling
- Embedding in Jupyter notebooks for interactive workflows

Example Outputs

- Show original abstract + ChatGPT summary
- Compare with SciSummary version
- Highlight differences in tone, clarity, and detail

Word embeddings: properties

- Relationships between words correspond to difference between vectors.



$$W(\text{“woman”}) - W(\text{“man”}) \simeq W(\text{“aunt”}) - W(\text{“uncle”})$$

$$W(\text{“woman”}) - W(\text{“man”}) \simeq W(\text{“queen”}) - W(\text{“king”})$$

http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/

Word embeddings: questions

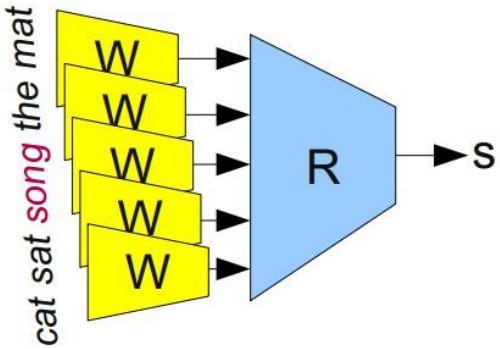
- How big should the embedding space be?
  - Trade-offs like any other machine learning problem – greater capacity versus efficiency and overfitting.
- How do we find W?
  - Often as part of a prediction or classification task involving neighboring words.

Learning word embeddings

- First attempt:
  - Input data is sets of 5 words from a meaningful sentence. E.g., “one of the best places”. Modify half of them by replacing middle word with a random word. “one of function best places”
  - W is a map (depending on parameters, Q) from words to 50 dim’l vectors. E.g., a look-up table or an RNN.
  - Feed 5 embeddings into a module R to determine ‘valid’ or ‘invalid’
  - Optimize over Q to predict better

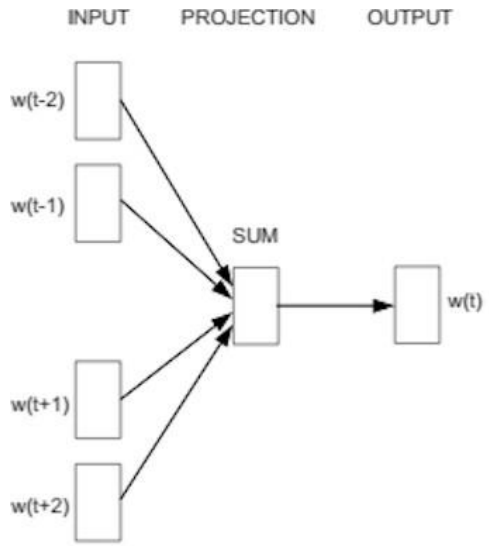
https://arxiv.org/ftp/arxiv/papers/1102/1102.1808.pdf

http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/

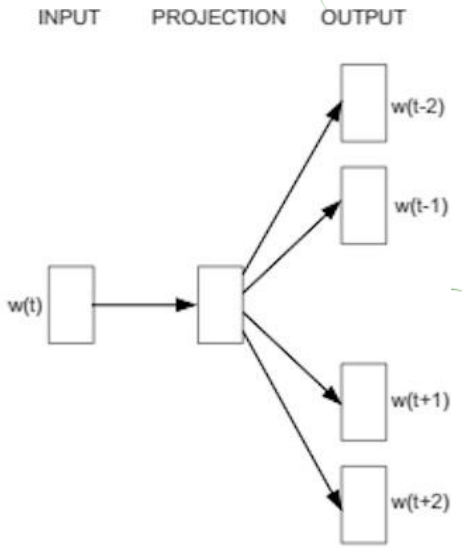


word2vec

- Predict words using context
- Two versions: CBOW (continuous bag of words) and Skip-gram



CBOW

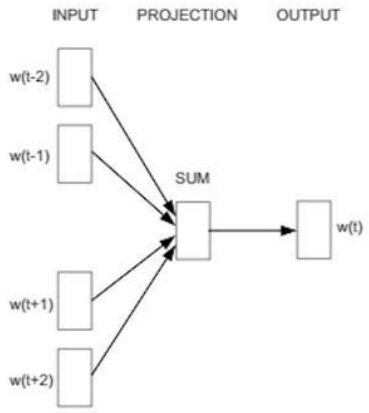


Skip-gram

https://skymind.ai/wiki/word2vec

CBOW

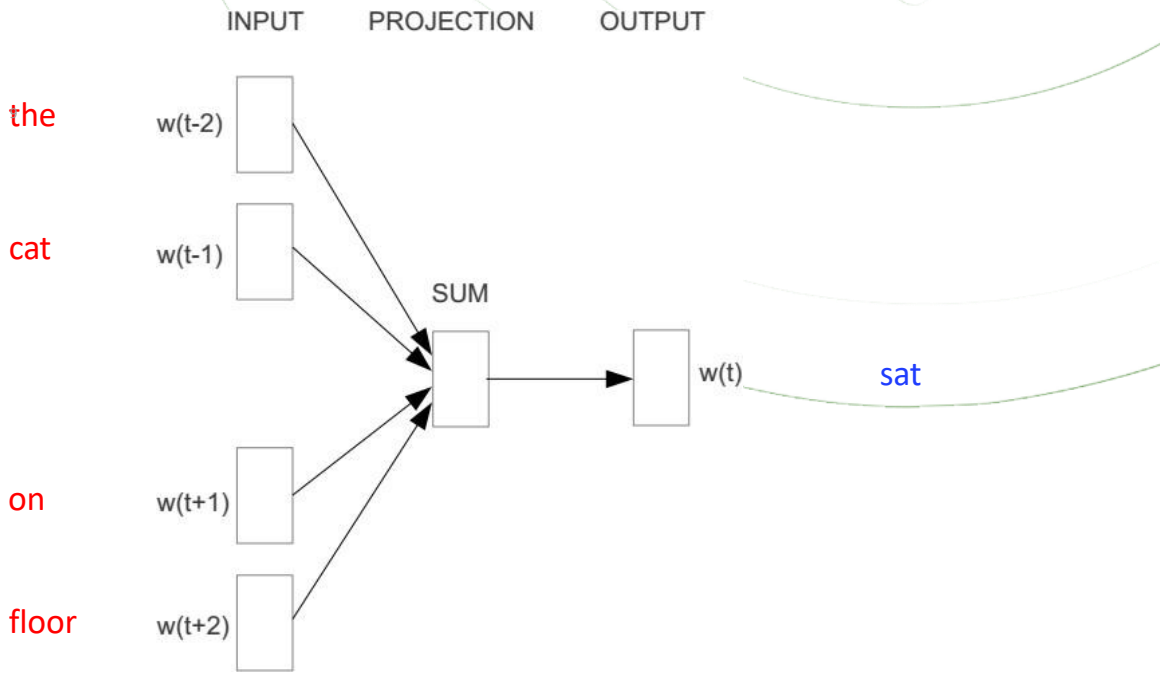
- Bag of words
  - Gets rid of word order. Used in discrete case using counts of words that appear.
- CBOW
  - Takes vector embeddings of n words before target and n words after and adds them (as vectors).
  - Also removes word order, but the vector sum is meaningful enough to deduce missing word.



CBOW

Word2vec – Continuous Bag of Word

- E.g. “The cat sat on floor”
  - Window size = 2



www.cs.ucr.edu/~vagelis/classes/CS242/slides/word2vec.pptx

Word Analogies

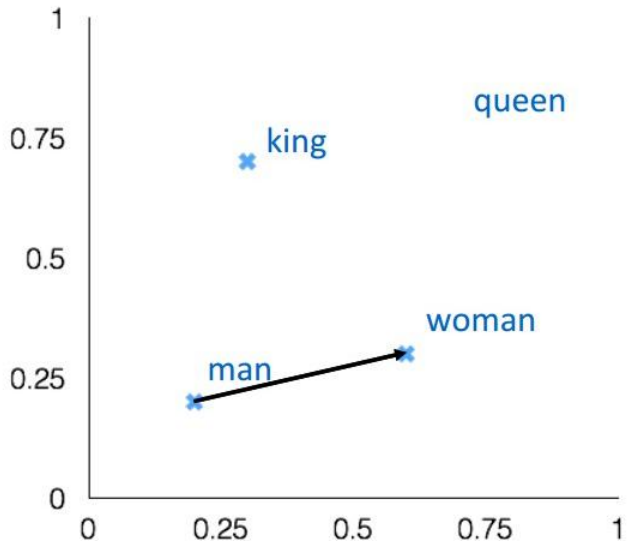
Test for linear relationships, examined by Mikolov et al. (2014)

a:b :: c:?

$$d = \arg\max_x \frac{(w_b - w_a + w_c)^T w_x}{||w_b - w_a + w_c||}$$

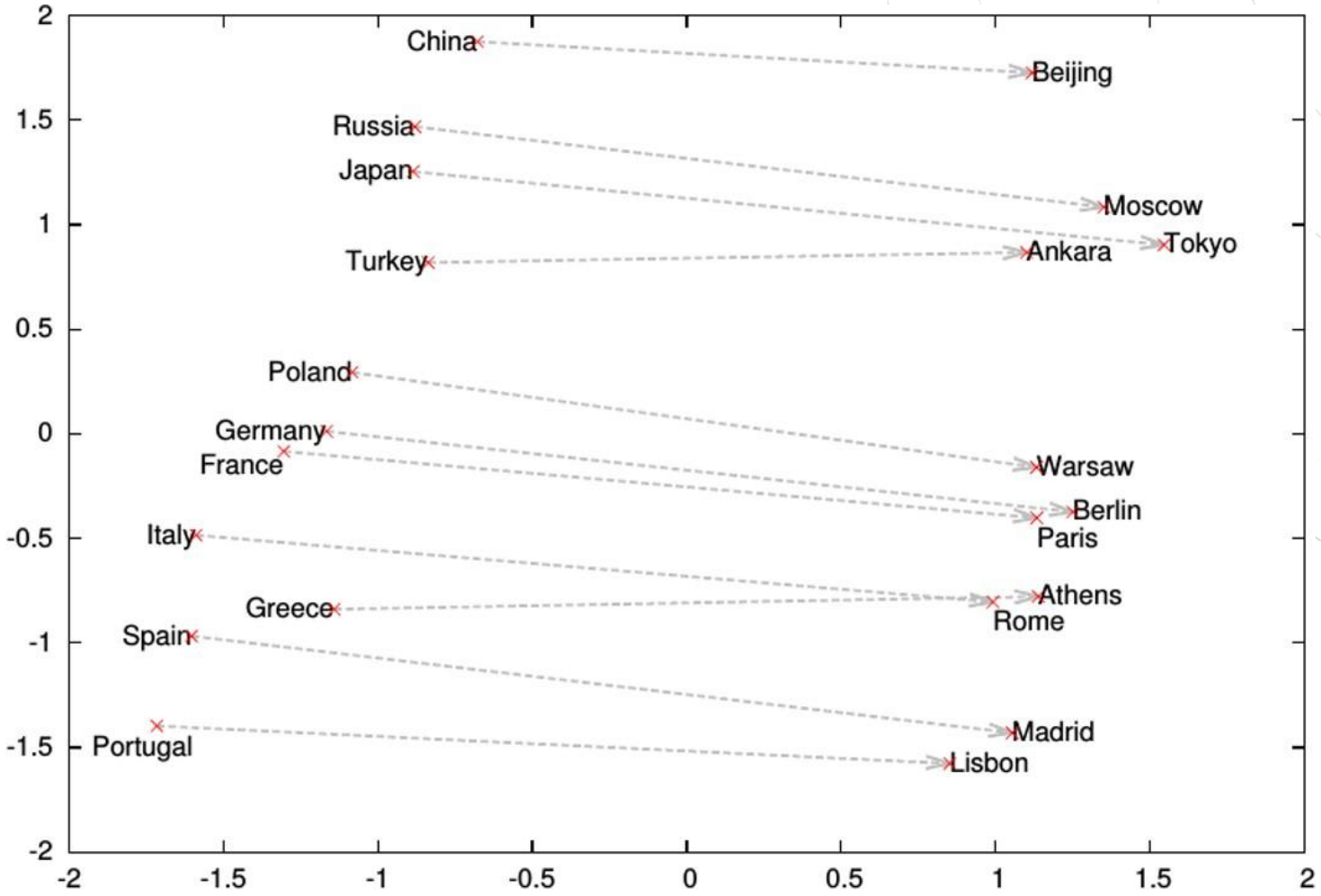
man:woman :: king:?

+	king	[0.30 0.70]
-	man	[0.20 0.20]
+	woman	[0.60 0.30]
	queen	[0.70 0.80]



www.cs.ucr.edu/~vagelis/classes/CS242/slides/word2vec.pptx

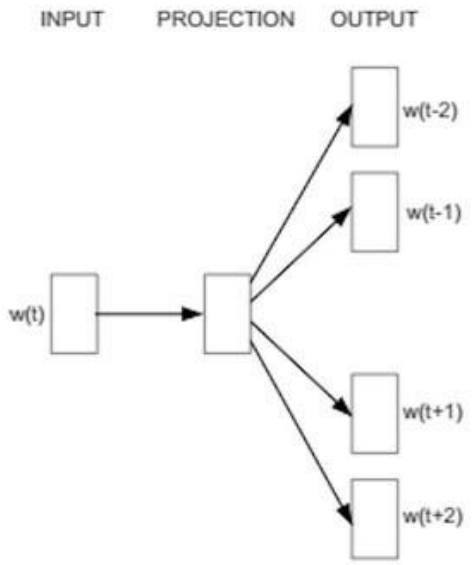
Word analogies



www.cs.ucr.edu/~vagelis/classes/CS242/slides/word2vec.pptx

Skip gram

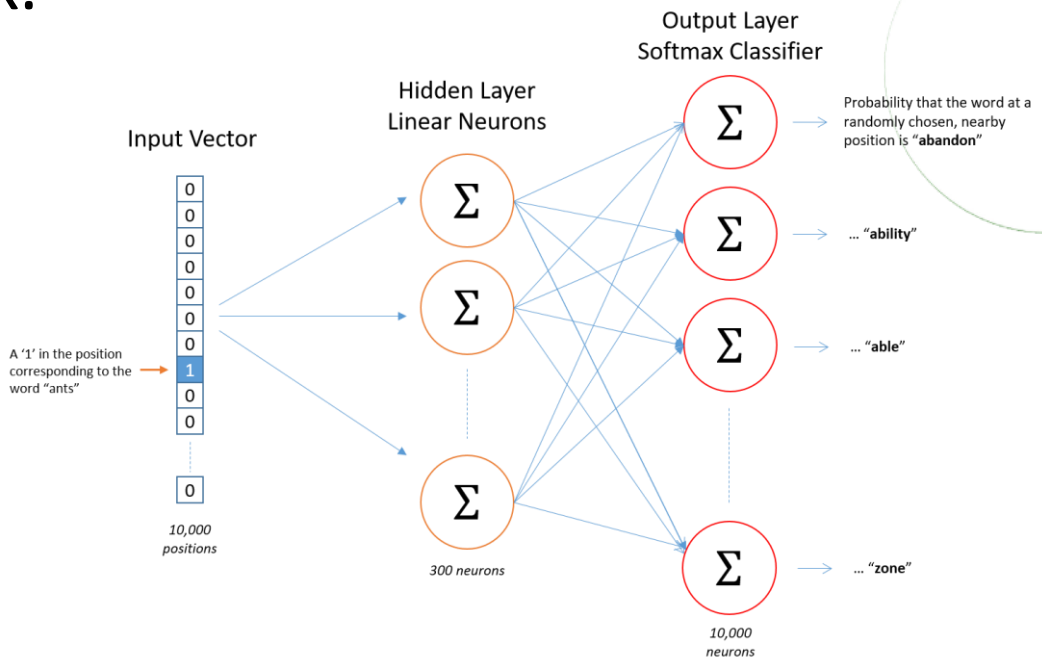
- Skip gram – alternative to CBOW
  - Start with a single word embedding and try to predict the surrounding words.
  - Much less well-defined problem, but works better in practice (scales better).



Skip-gram

Skip gram

- Map from centre word to probability on surrounding words. One input/output unit below.
- There is no activation function on the hidden layer neurons, but the output neurons use softmax.

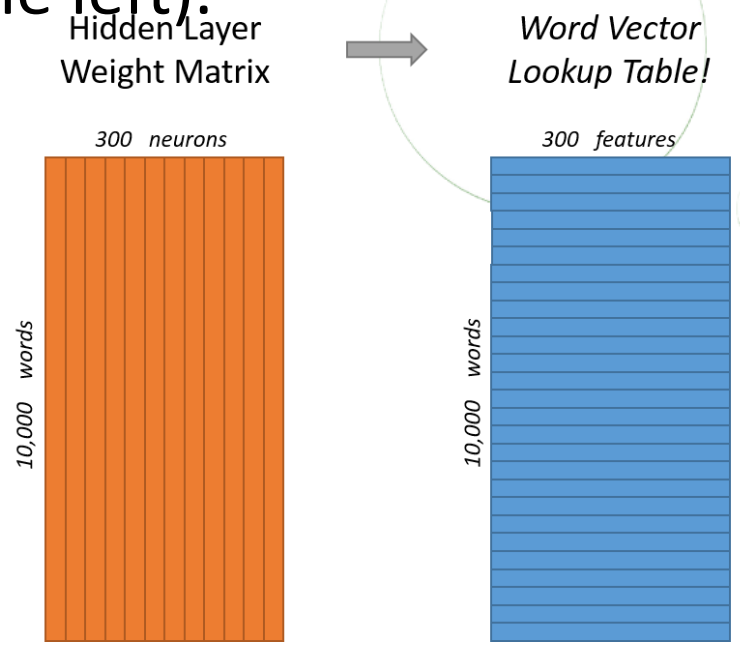


http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Skip gram example

- Vocabulary of 10,000 words.
- Embedding vectors with 300 features.
- So the hidden layer is going to be represented by a weight matrix with 10,000 rows (multiply by vector on the left).

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$



http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Skip gram/CBOW intuition

- Similar “contexts” (that is, what words are likely to appear around them), lead to similar embeddings for two words.
- One way for the network to output similar context predictions for these two words is if the word vectors are similar. So, if two words have similar contexts, then the network is motivated to learn similar word vectors for these two words!

http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

word2vec

- word2vec, or skip-gram, is an algorithm for training real-valued vectors to represent each word.
- If word $w1$ is represented by vector $\vec{v}1 = v11, \ldots, v1D$, we say that $\vec{v}1$ is the D-dimensional embedding of word $w1$.
- The general area of vector semantics (representthe meaning of a word as a vector) goes back to the 1950s, in the field of information retrieval (more about that in the next lecture).
- word2vec is an algorithm for learning those vectors using a one-layer neural network, in such a way that similar words are close together in the vector space.

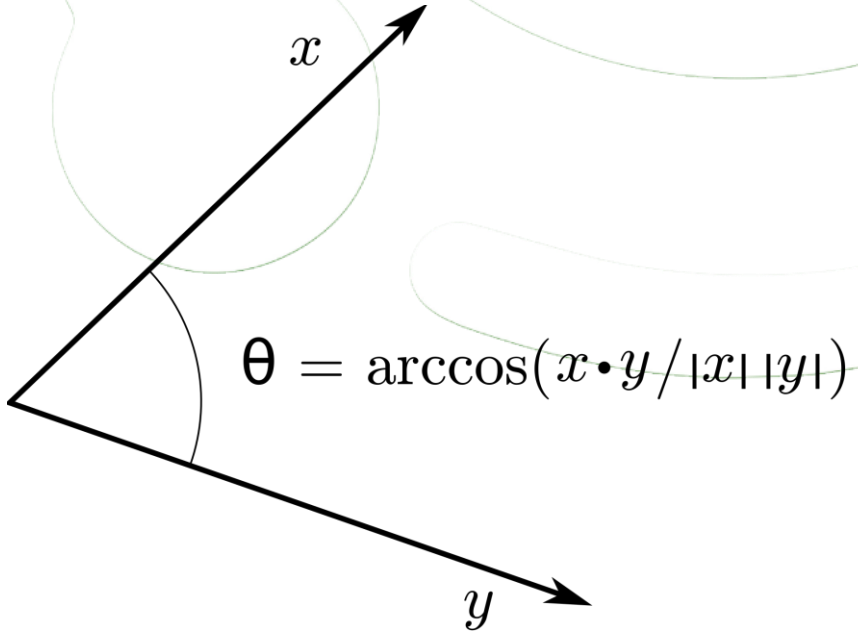
cosine similarity

- If words $w1$ and $w2$ are similar, $w1$is represented by vector $\vec{v}1$, and $w2$ by vector $\vec{v}2$, then the angle between the two vectors should be small.
- Angle between two vectors can be measured by their dot product

$$\cos\theta = \frac{\vec{v}_1 \,) \vec{v}_2}{|\vec{v}_1||\vec{v}_2|}$$

where

$$\vec{v}_1 \,) \vec{v}_2 = \sum_{d=1}^{D} v_{1d} v_{2d}\,, \qquad |\vec{v}_1| = \sqrt{\sum_{d=1}^{D} v_{1d}^2}$$



By BenFrantzDale at the English Wikipedia, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=49972362

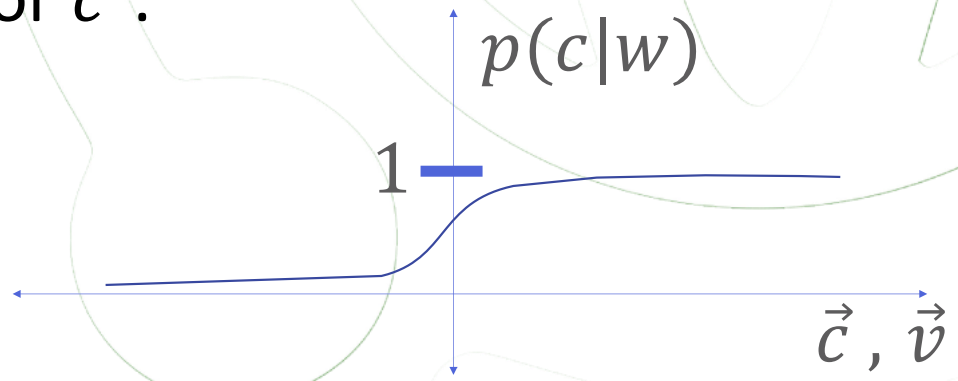
Word2vec: context probability

- The key innovation of word2vec is the idea of representing similarity as the probability that words $w1$ and $w2$ could occur in the same context, and of estimating the probability using a sigmoid.
- Consider the "…hot although iced coffee is a popular…". Define the target word to be w =coffee.
- Define the context words $c-3$ =hot, $c-2$ =although, …, $c3$ =popular. Use a naïve Bayes model of the context probability:

$$p(c_{-3}, \ldots, c_3 | w) = \prod_{\substack{i=-3 \\ i \neq 0}} p(c_i | w)$$

word2vec: context probability

- Now suppose we want to embed w =coffee with a vector $\vec{v}$.
- …and we want to embed c–3 =hot with a vector $\vec{c}$.
- Define the probability that “hot” occurs within
- +/-N words of “coffee” to be just a sigmoid:



$$p(c|w) = \frac{1}{1 + e^{-c \cdot \vec{v}}}$$

word2vec: training

- We train the neural network by listing, as positive examples, the words that occur in the context of "w =coffee," e.g.,
  - $\mathcal{D}+(w)$ = {hot, although, iced, moderate, the, hot, consumption, … }
- Create a negative database by selecting words at random from the vocabulary, each word in proportion to its frequency in the whole dataset:
  - $\mathcal{D}-(w)$ ={aardvark, dog, gazebo, the, precipitates, … }

Word2vec shortcomings

- Problem: 10,000 words and 300 dim embedding gives a large parameter space to learn. And 10K words is minimal for real applications.
- Slow to train, and need lots of data, particularly to learn uncommon words.

Word2vec improvements: word pairs and phrases

- Idea: Treat common word pairs or phrases as single “words.”
  - E.g., Boston Globe (newspaper) is different from Boston and Globe separately. Embed Boston Globe as a single word/phrase.
- Method: make phrases out of words which occur together often relative to the number of individual occurrences. Prefer phrases made of infrequent words in order to avoid making phrases out of common words like “and the” or “this is”.
- Pros/cons: Increases vocabulary size but decreases training expense.
- Results: Led to 3 million “words” trained on 100 billion words from a Google News dataset.

Word2vec improvements: subsample frequent words

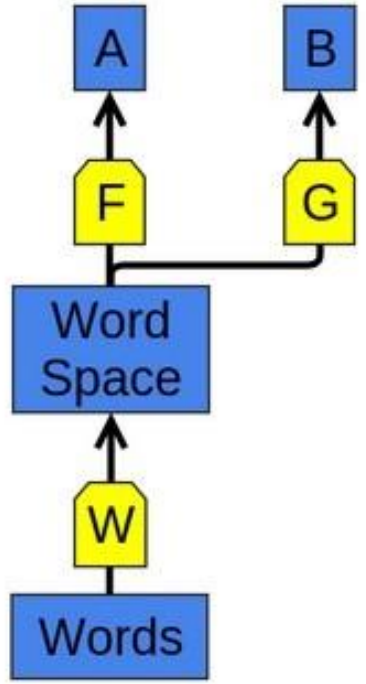
- Idea: Subsample frequent words to decrease the number of training examples.
  - The probability that we cut the word is related to the word's frequency. More common words are cut more.
  - Uncommon words (anything < 0.26% of total words) are kept
  - E.g., remove some occurrences of "the."
- Method: For each word, cut the word with probability related to the word's frequency.
- Benefits: If we have a window size of 10, and we remove a specific instance of "the" from our text:
  - As we train on the remaining words, "the" will not appear in any of their context windows.

Word2vec improvements: selective updates

- Idea: Use “Negative Sampling”, which causes each training sample to update only a small percentage of the model’s weights.
- Observation: A “correct output” of the network is a one-hot vector. That is, one neuron should output a 1, and all of the other thousands of output neurons to output a 0.
- Method: With negative sampling, randomly select just a small number of “negative” words (let’s say 5) to update the weights for. (In this context, a “negative” word is one for which we want the network to output a 0 for). We will also still update the weights for our “positive” word.

Word embedding applications

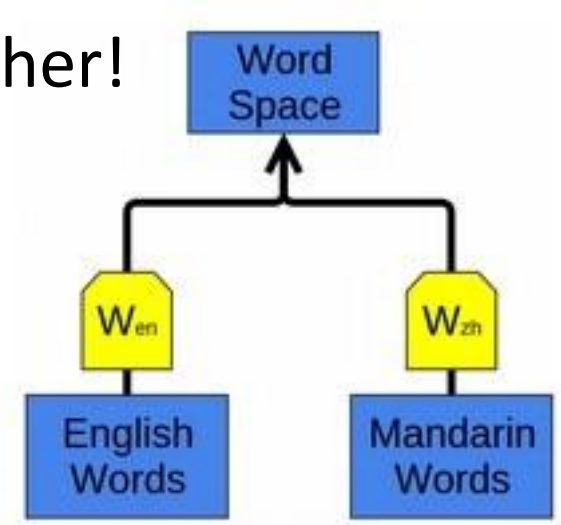
- The use of word representations… has become a key "secret sauce" for the success of many NLP systems in recent years, across tasks including named entity recognition, part-of-speech tagging, parsing, and semantic role labelling. (Luong et al. (2013))
- Learning a good representation on a task A and then using it on a task B is one of the major tricks in the Deep Learning toolbox.
  - Pretraining, transfer learning, and multi-task learning.
  - Can allow the representation to learn from more than one kind of data.



W and F learn to perform task A. Later, G can learn to perform B based on W.

Word embedding applications

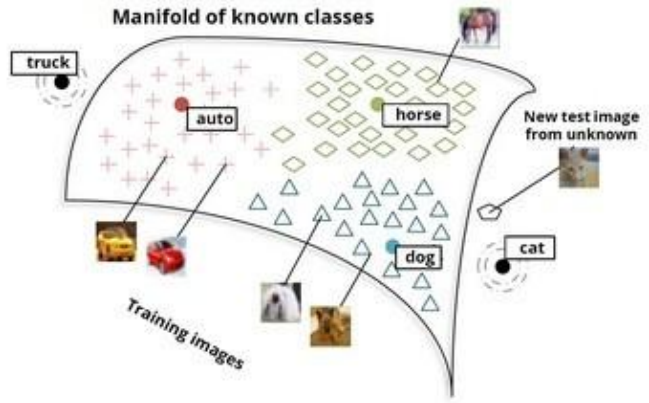
- Can learn to map multiple kinds of data into a single representation.
- E.g., bilingual English and Mandarin Chinese word-embedding as in Socher et al. (2013a).
- Embed as above, but words that are known as close translations should be close together.
- Words we didn't know were translations end up close together!
- Structures of two languages get pulled into alignment.



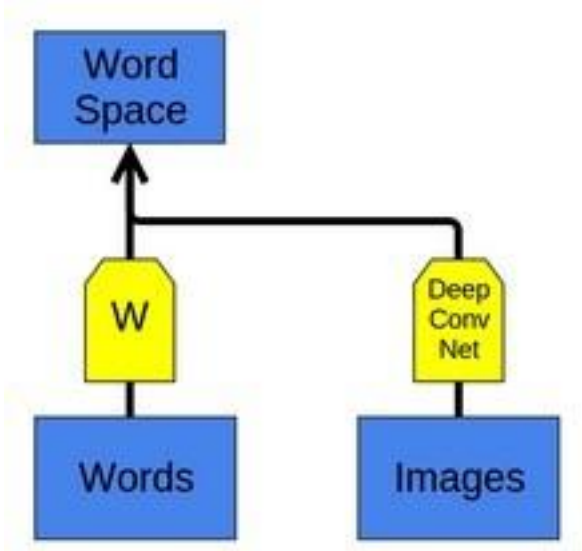
http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/

Word embedding applications

- Can apply to get a joint embedding of words and images or other multi-modal data sets.
- New classes map near similar existing classes: e.g., if ‘cat’ is unknown, cat images map near dog.



(Socher *et al.* (2013b))



http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/

Key Benefits Recap

- Save time scanning papers
- Improve comprehension for non-native readers
- Make science more accessible to collaborators and the public

Resources & Tools

- Tools: scisummary.com, paperqa.xyz
- Plugins: ScholarAI (ChatGPT), Scite
- Datasets: arXiv, PubMed, Semantic Scholar Open Research Corpus

Q&A and Feedback

- Prompt
  - “What are the pros and cons of using LLMs in your research field?”
- Invite discussion
- Optional: quick survey poll (Mentimeter or Slido)

Final Activity: Case Studies & Group Work

- Group work: each team chooses a case (e.g., water monitoring, deforestation, urban heat islands)
- Define the problem, select relevant data and AI approach
- Share ideas with the class (5-min lightning pitch per group)

THANKS!

Francesco Iarlori

https://uk.linkedin.com/in/thefrankie/

@thefrankie

Francesco@Iarlori.com

IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-Mission 4 “Education and Research” - Component 2: “From research to business” - Investment 3.1: “Fund for the realisation of an integrated system of research and innovation infrastructures”

Finanziato dall'Unione europea NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani

Teachable Machine: Bird Species Recognition - Ex09

- Topic: Biodiversity Monitoring
- Tool: Google Teachable Machine
- Exercise:
  - Students upload or record audio clips (e.g. birdsong samples from Xeno-Canto or AudioSet).
  - Train a basic model to recognize different bird species.
  - Discuss performance and real-world limitations.
- Goal: Show the simplicity of model creation and limitations of generalization with small datasets.

Google Earth Engine: Land Cover Classification - Ex10

- Topic: Remote Sensing
- Tool: Google Earth Engine
- Exercise:
  - Use a public script (or provide one) that classifies land cover (forest, urban, water) in a chosen region.
  - Adjust parameters and visualize classification results.
  - Optional: Compare results from different years.
- Goal: Understand how satellite imagery and classification algorithms support environmental monitoring.

Colab Notebook: Air Quality Anomaly Detection - Ex11

- Topic: Big Environmental Data
- Tool: Google Colab
- Exercise:
  - Load a subset of OpenAQ data (PM2.5 or NO2 from Milan or another local city).
  - Clean the dataset using pandas.
  - Use scikit-learn to detect anomalies with Isolation Forest or Z-score method.
  - Visualize results with matplotlib.
- Goal: Learn how to process time-series data and detect anomalies using ML.

Time-Series Forecasting with RNNs (Opt. Adv.)

- Topic: Climate Modeling
- Tool: Colab + TensorFlow/Keras
- Exercise:
  - Load simplified climate data (e.g., monthly temperature anomalies).
  - Train an LSTM model to predict future values.
  - Evaluate MAE and plot prediction vs actual.
- Goal: Understand how deep learning can model temporal patterns in climate data.

LLM Comparison: Literature Summarization

- Topic: LLMs in Research
- Tools: ChatGPT, Elicit, SciSummary
- Exercise:
  - Provide each group with a paragraph from a scientific abstract or paper (environment-related).
  - Ask them to summarize using ChatGPT or another LLM.
  - Compare the quality, tone, and usefulness of summaries across tools.
- Goal: Explore how LLMs can speed up literature review and knowledge extraction.