Show Don't Tell: Designing Effective AI Demos for K-12 Students

David S. Touretzky
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA USA

AIDE 2021, Beijing, November 16-17, 2021

AI4K12.org

The AI4K12 Initiative (AI4K12.org), a joint project of:

AAAI (Association for the Advancement of Artificial Intelligence)

CSTA (Computer Science Teachers Association)

With funding from National Science Foundation ITEST Program (DRL-1846073)

Carnegie Mellon University
School of Computer Science

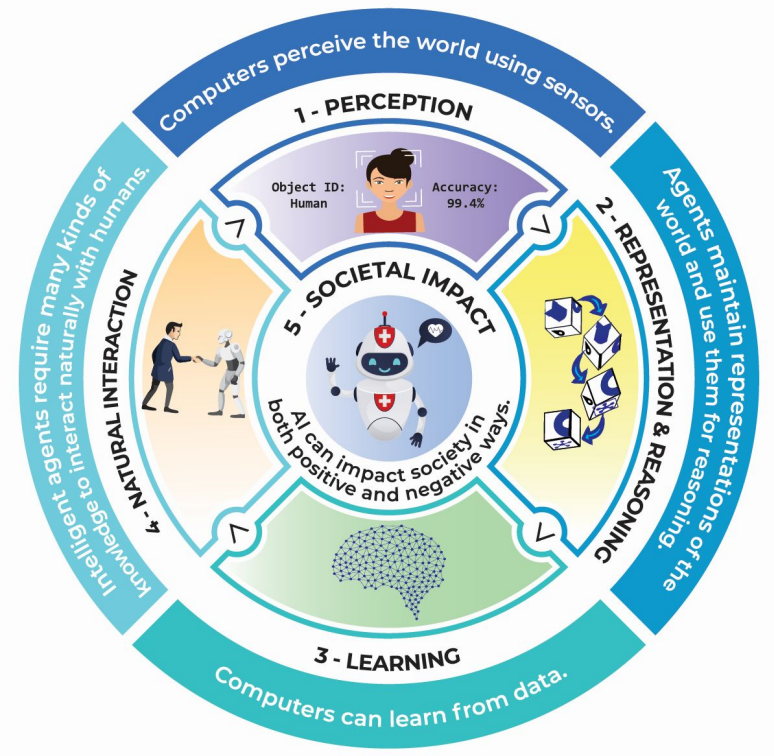
AI4K12.org Mission

- Develop national guidelines for teaching AI in K-12
  - Modeled after the CSTA standards for computing education.
  - Four grade bands: K-2, 3-5, 6-8, and 9-12
  - What should students know?
  - What should students be able to do?
- Develop a curated AI resource directory for K-12 teachers

  ⇒ Including online demos
- Foster a community of K-12 AI educators, researchers, and resource developers

Five Big Ideas in AI

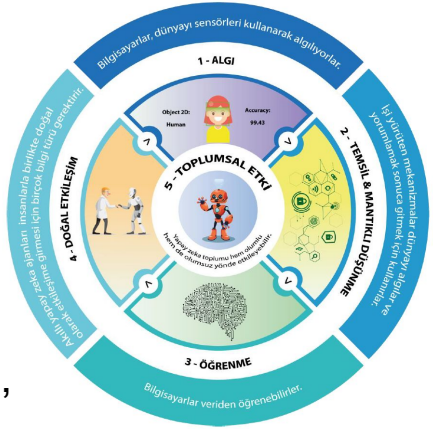
1. **Perception:** Computers perceive the world using sensors.
2. **Representation and reasoning:** Agents maintain representations of the world and use them for reasoning.
3. **Learning:** Computers can learn from data.
4. **Natural interaction:** Intelligent agents require many kinds of information to interact naturally with humans.
5. **Societal impact:** AI can impact society in both positive and negative ways.



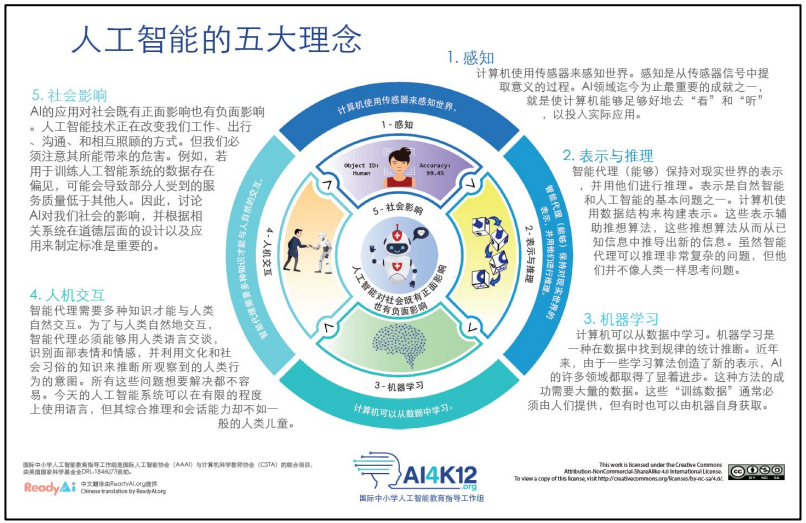
Adoption of the Big Ideas

- Now being adopted by curriculum developers in the US and elsewhere.
- Translations available in 16 languages including Chinese, Korean, Japanese, Spanish, Portugese, Hebrew, Arabic, Hindi, and Thai.

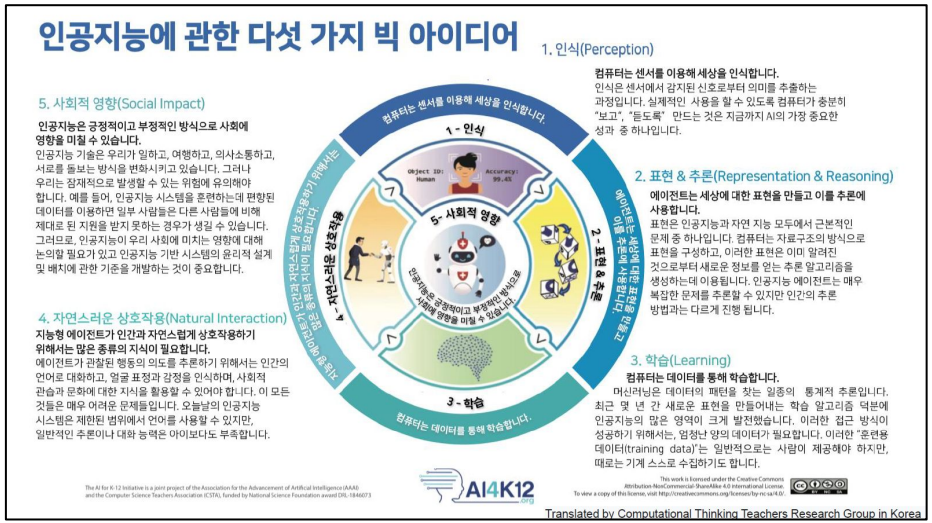
Turkish



Chinese



Korean

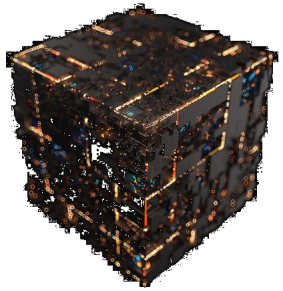


Why Demos Matter

- AI education exposes students to highly abstract concepts:
  - Knowledge representation
  - Reasoning algorithms
  - Statistical models
  - Deep neural networks
  - The nature of language

- Good interactive demos can make these ideas approachable even by young students, despite:
  - Little or no programming experience
  - Limited mathematical sophistication

Black Box vs. Glass Box Demos



Black box demos:

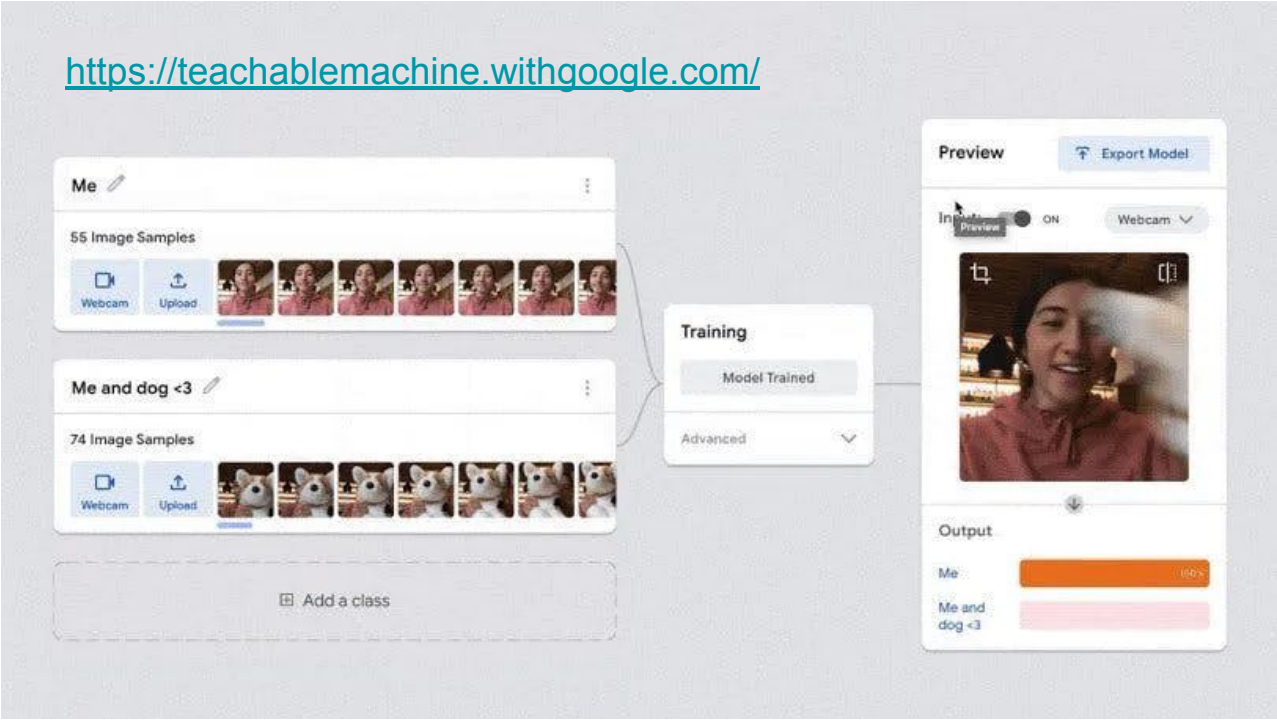
- Demonstrate some engaging application of AI, possibly quite complex.
- Offer no view of their internal workings.
- Provide limited insight into the strengths/limitations of the approach.

Glass box demos:

- Reveal their workings, providing a view “under the hood”.
- Use simpler tasks so that the computations are understandable.

Classic Black Box Demo: Google's Teachable Machine

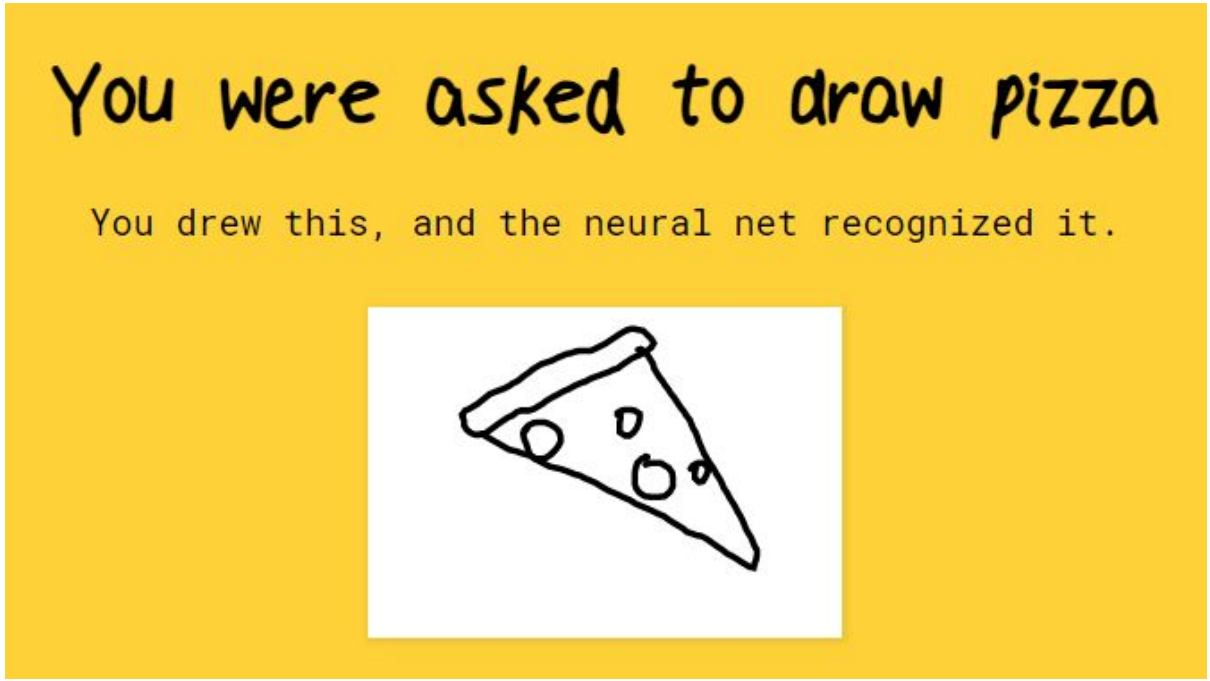
Train a classifier using your webcam or microphone.



- Train a classifier in as little as 2 minutes.
- Good for all ages.
- Export trained classifier to Scratch, Python or JavaScript.
- Uses transfer learning with a deep neural net, but users can't see that.

Classic Black Box Demo: Google's Quick Draw

Draw an object for a neural net to recognize. https://quickdraw.withgoogle.com/

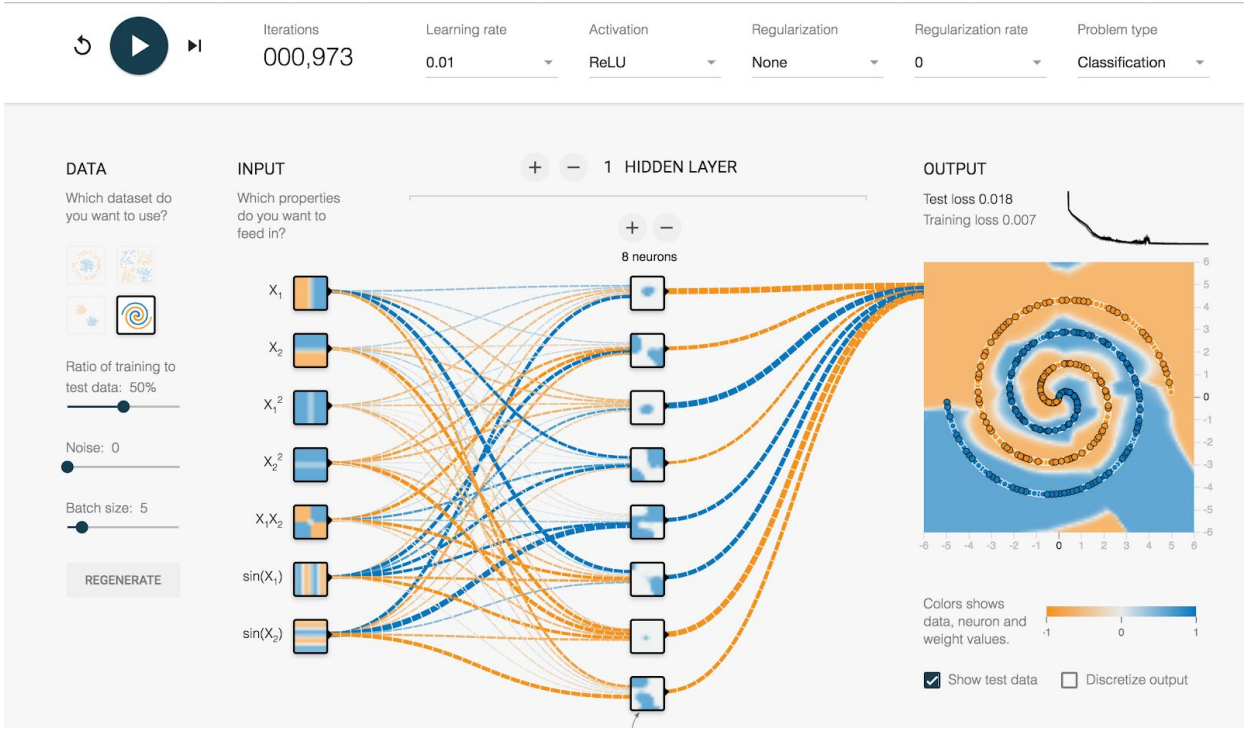


- Game-ified demo: try to draw 6 objects the computer chooses.
- Neural net makes guesses in real time as you draw.
- 20 seconds per drawing.
- Your drawings provide additional training data to improve the game.
- But how does it work???

https://playground.tensorflow.org/

Glass Box Demo: TensorFlow Playground

Train a neural net classifier using backpropagation.



- Complete transparency: units, connections, weights, responses, learning parameters.
- Watch the network evolve in real time during training.
- Weights are user-modifiable.
- Suitable for advanced high school students.

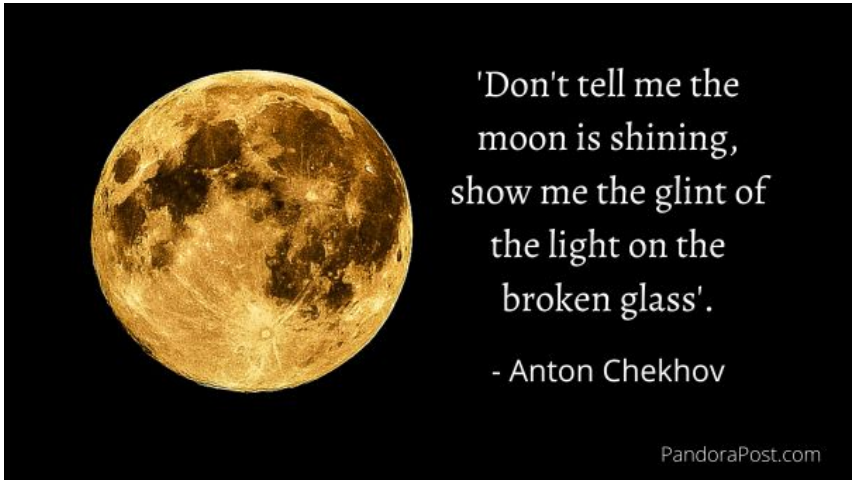
Importance of Running in the Browser

- Schools use a variety of platforms: MacOS, Windows, iPads, Chromebooks.
- Many schools lock down their computers to prevent malware attacks.
- Installing new software can only be done by IT staff and may require long lead times: as much as a month or more.
- A web browser like Chrome or Firefox is common across platforms, and can run JavaScript applications without needed to download and install anything.

Four Rules for Creating Effective Interactive Demos

(1) Show Don't Tell

- Visualization aids understanding.
- Use graphics to illustrate the demo's representations and processing.
- *"A picture is worth a thousand words."*



(2) Be intriguing at first glance

- Choose defaults that generate interesting behavior at the start of the demo.
- Don’t make the user work to get the first reward.

(3) Don't show too much too soon

- Beginners will be intimidated by too much complexity.
- Demos with complex options and controls should hide these from the first-time user.
- Reveal more depth as the user is ready for it.

(4) Build in support for teachers

- Include Tutorial Materials to explain how the demo works and the technology behind it.

- Include a list of Experiments students can perform using the demo, and explain what they should notice in the results.

SpeechDemo

https://www.cs.cmu.edu/~dst/SpeechDemo

Speech Recognition Demo

Speak into your microphone; see the results below.

Click here for experiments to try.

pause | English (US) | read back

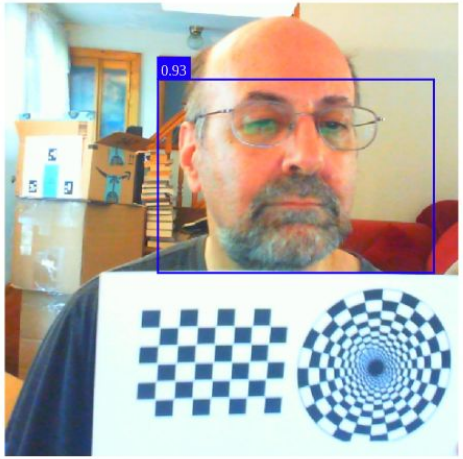
demos with complex controls and options should hide these from beginning users
demos with complex controls an option should hide these from beginning users
demos with complex controls in options should hide these from beginning users
demos with complex controls and option should hide these from beginning users
temos with complex controls and options should hide these from beginning users

MIC ON

FaceDemo

https://www.cs.cmu.edu/~dst/FaceDemo

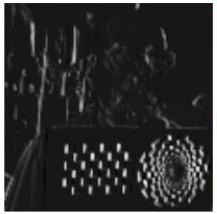
FaceDemo: Deep Neural Network for Face Detection



Convolutional Layer : 1

R +
R -
G +
G -
B +
B -

Kernel Select
2 - Green patch detector



R +
R -
G +
G -
B +
B -

Kernel Select
9 - Vertical edge detector

R +
R -
G +
G -
B +
B -

Kernel Select
11 - Red patch detector

Convolutional Layer : 4

Kernel:
26

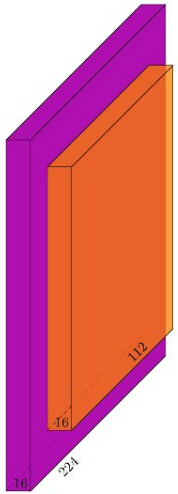
Kernel:
36

Kernel:
46

Kernel:
112

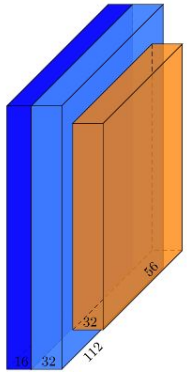
FaceDemo Tutorial: TinyYoloV2 Architecture

Input Image
R, G, B Channels
224×224×3



conv1: [3,3,3]/1 × 16
224×224×16

max1: [2,2,1]/2 × 16
112×112×16



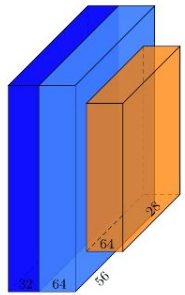
conv2: [3,3,1]/1 × 16
112×112×16

conv3: [1,1,16]/1 × 32
112×112×32

max2: [2,2,1]/2 × 32
56×56×32

Conv & Pool Key: kernelType [kw,hw,inchan]/stride × outchan
Layer Dimension Key: width × height × channels

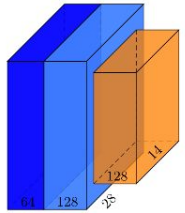
Purple: Full Convolution
Dark Blue: Depthwise Convolution
Light Blue: Pointwise Convolution
Orange: Max Pooling



conv4: [3,3,1]/1 × 32
56×56×32

conv5: [1,1,32]/1 × 64
56×56×64

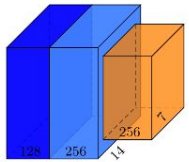
max3: [2,2,1]/2 × 64
28×28×64



conv6: [3,3,1]/1 × 64
28×28×64

conv7: [1,1,64]/1 × 128
28×28×128

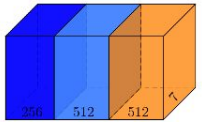
max4: [2,2,1]/2 × 128
14×14×128



conv8: [3,3,1]/1 × 128
14×14×128

conv9: [1,1,128]/1 × 256
14×14×256

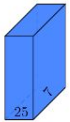
max5: [2,2,1]/2 × 256
7×7×256



conv10: [3,3,1]/1 × 256
7×7×256

conv11: [1,1,256]/1 × 512
7×7×512

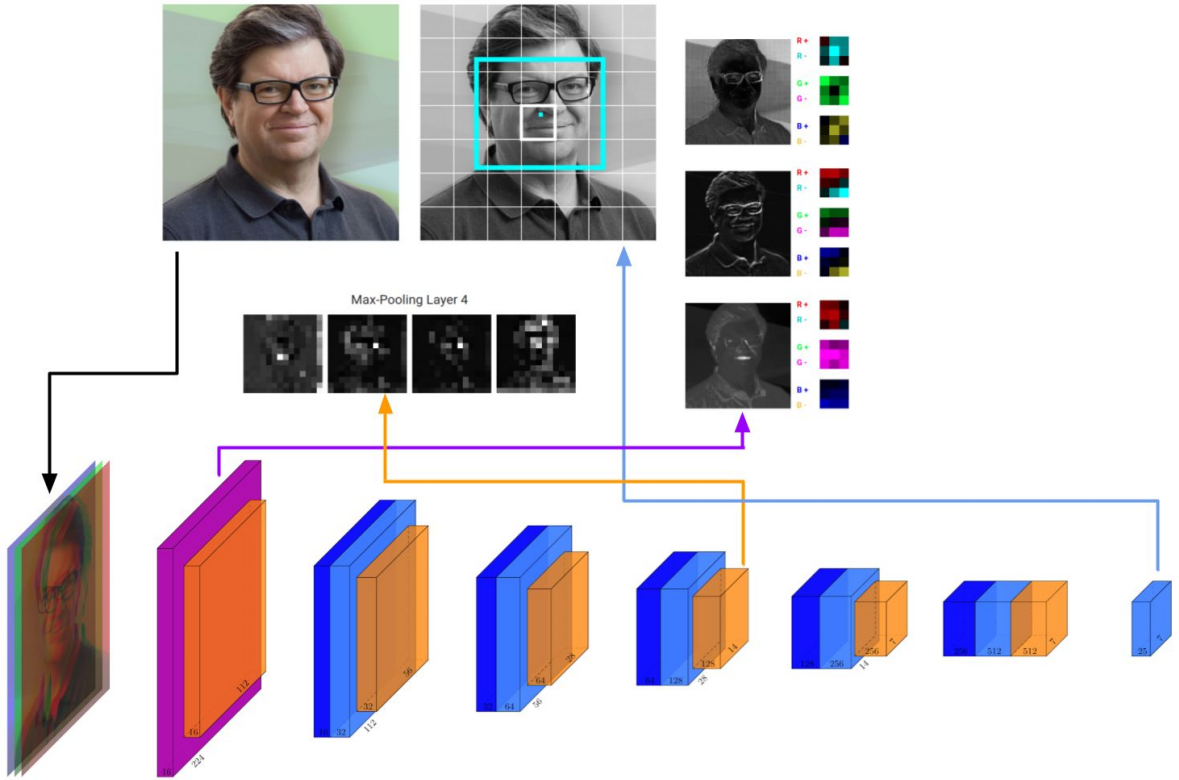
max6: [2,2,1]/1 × 512
7×7×512



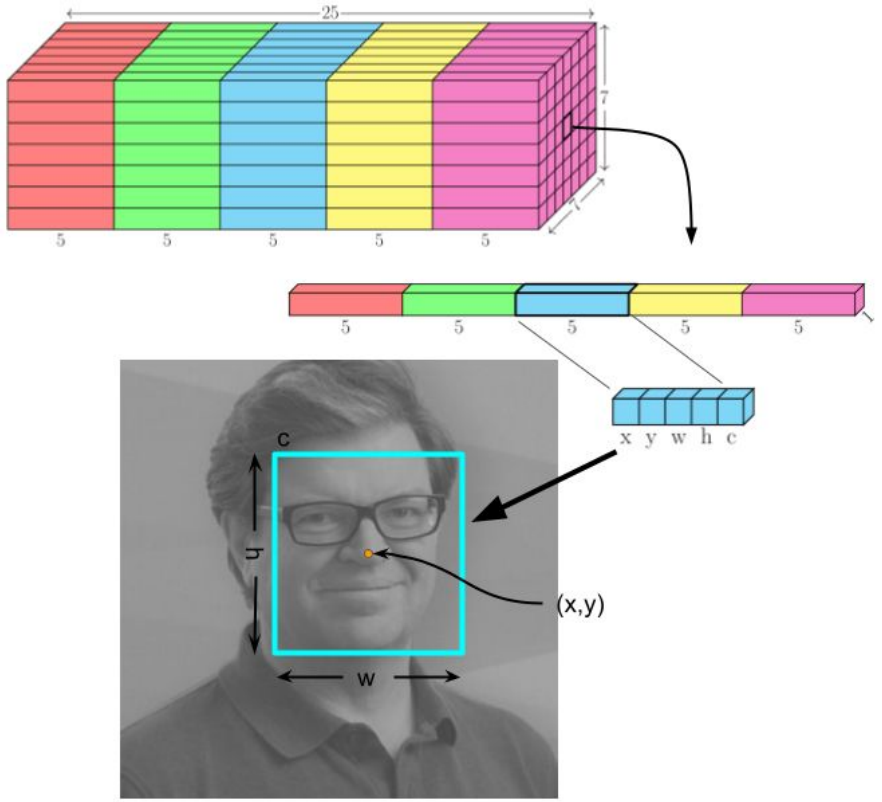
conv12: [1,1,512]/1 × 25
7×7×25

Layer depths are drawn with logarithmic scaling. Max pooling layer widths and heights are shrunk by $\frac{1}{4}$ instead of $\frac{1}{2}$.

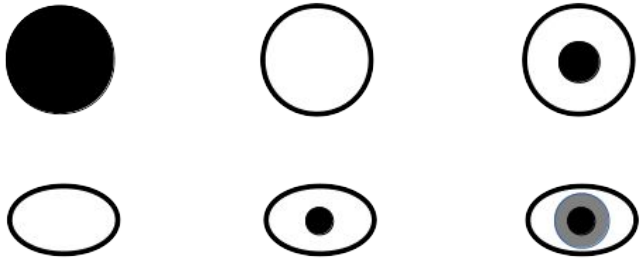
FaceDemoTutorial: Mapping the Network to the Display

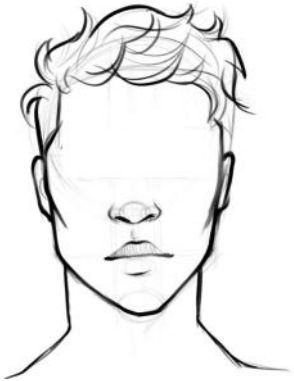


FaceDemo Tutorial: Extracting the Bounding Box



FaceDemo Experiments





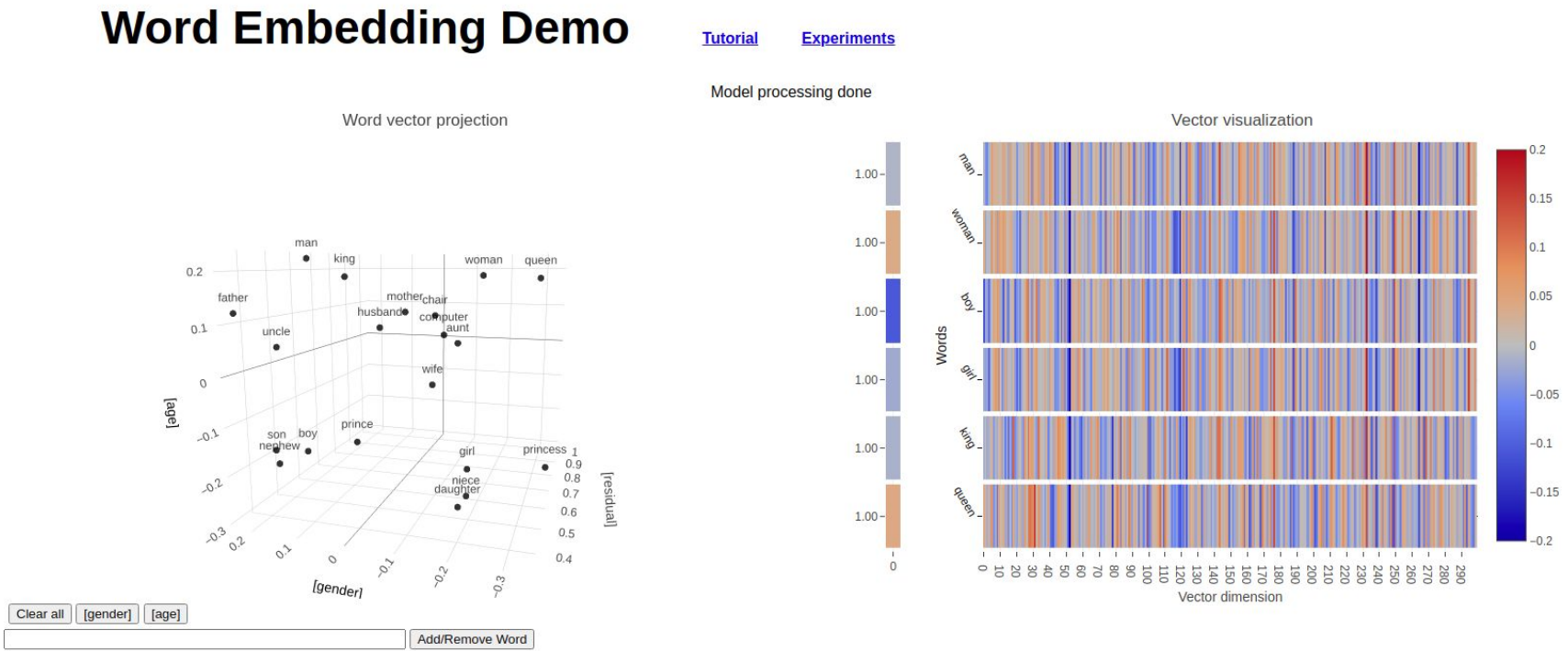
WordEmbeddingDemo

https://www.cs.cmu.edu/~dst/WordEmbeddingDemo

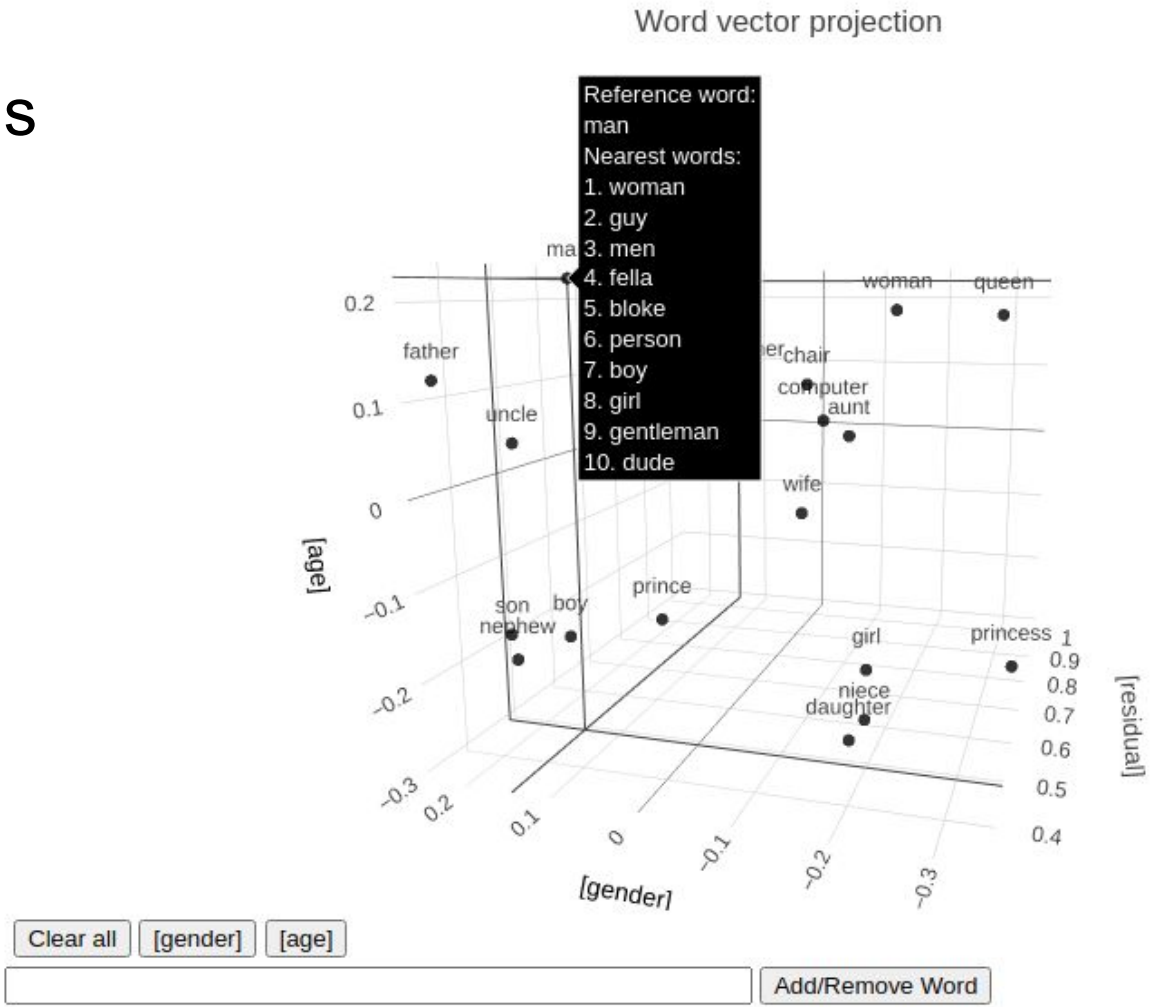
What are word embeddings?

- Vector representations of words.
- Created by neural net learning algorithms such as **word2vec**.
- Used as the input code for neural natural language processing applications, such as:
  - Transformer networks
  - Machine translation
  - Question answering
  - Text generation

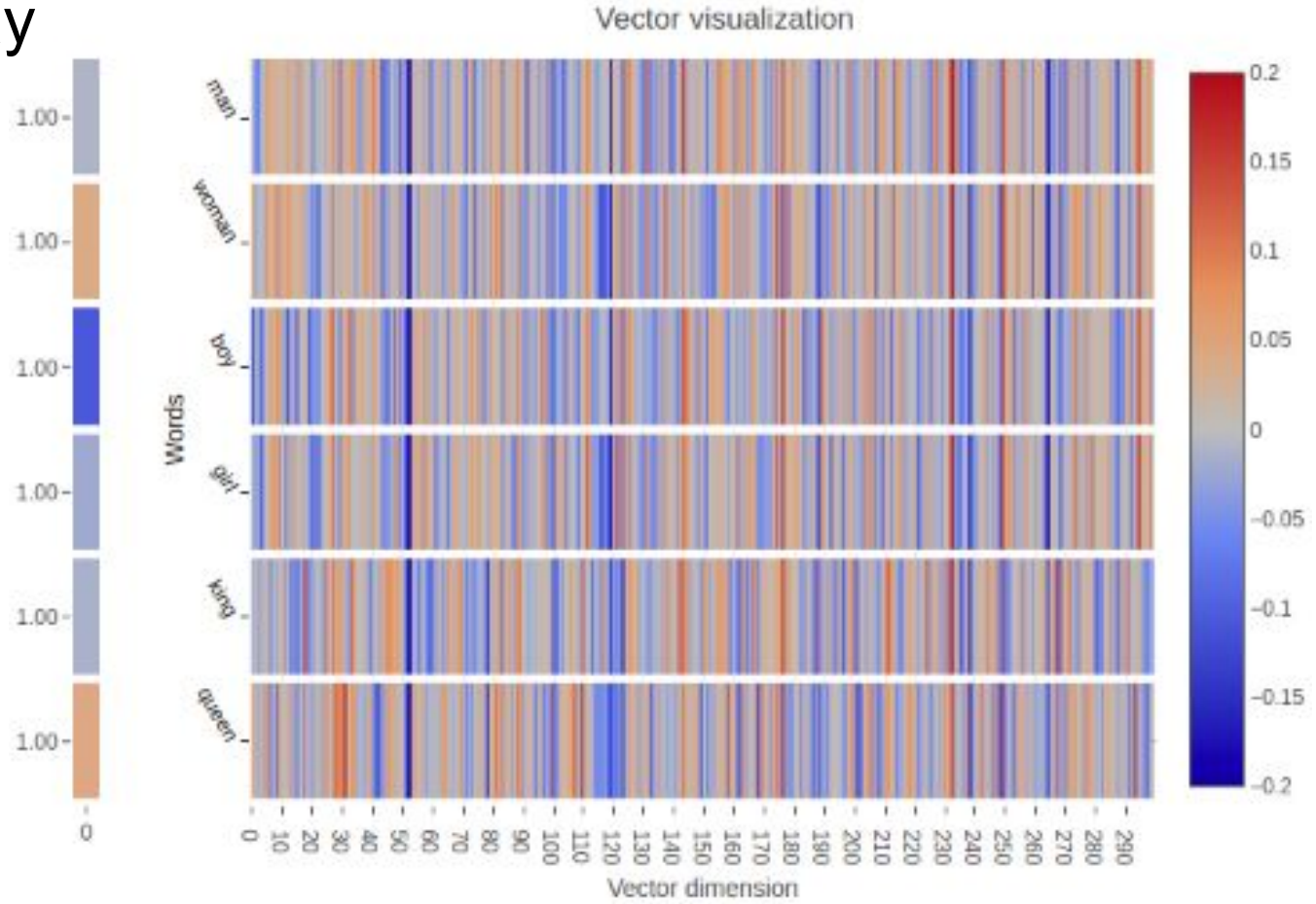
Graphical Exploration of Word Embeddings



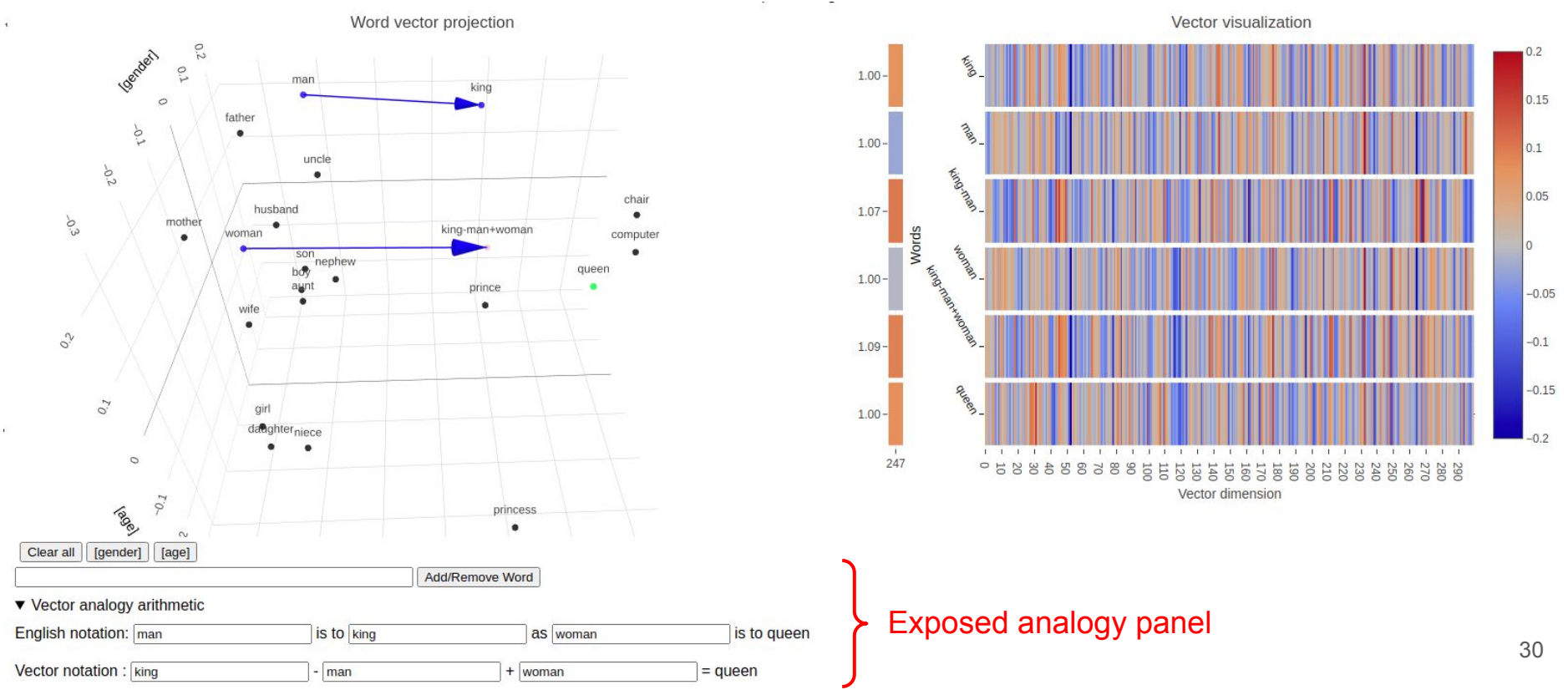
Closest words



Vector Display

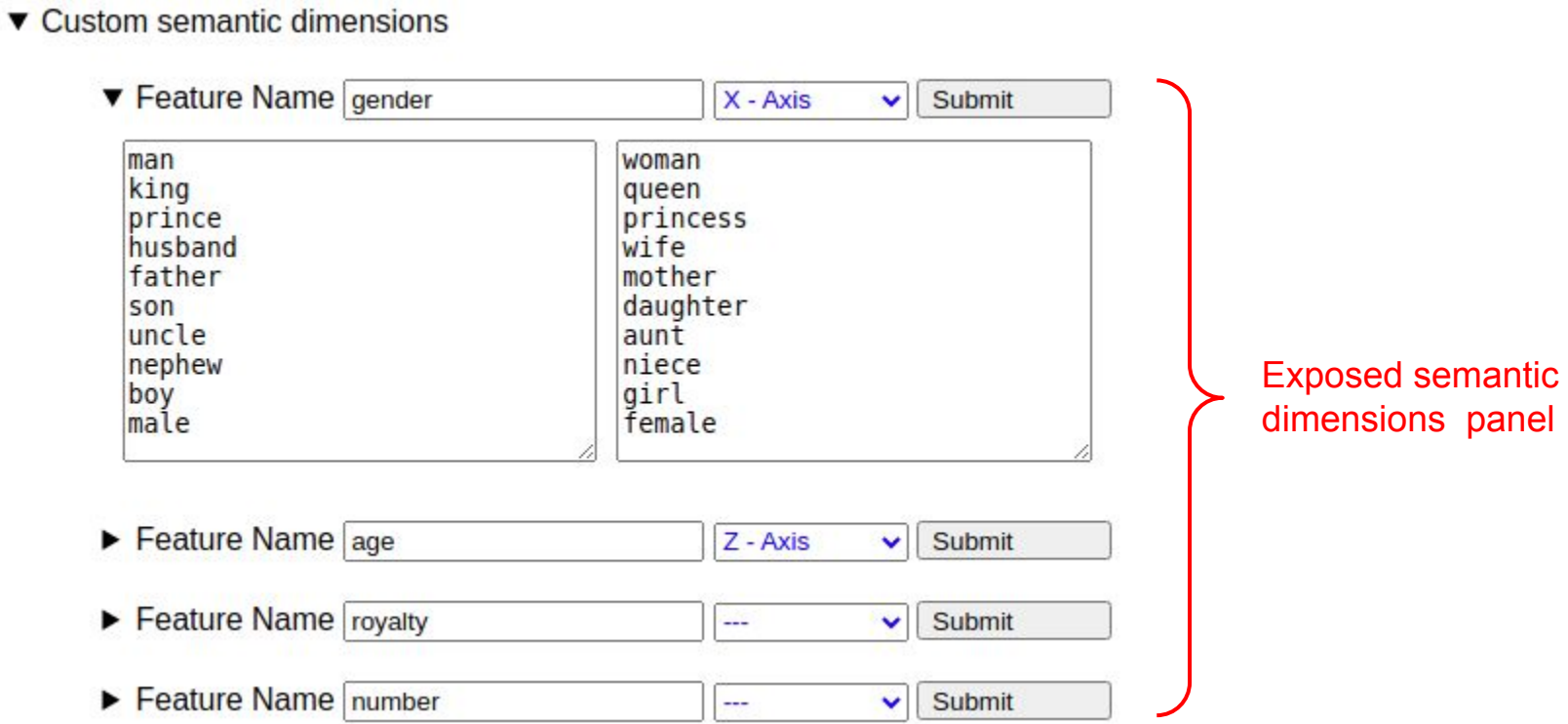


Analogy by vector arithmetic



Exposed analogy panel

Exposing the semantic dimension definitions



Summary

- We are seeing a world-wide explosion of interest in K-12 AI education.
- Interactive demos can play an important role in supporting K-12 learning.
  - Black box demos show off what AI can do.
  - Glass box demos offer a glimpse of how AI works.
- For accessibility, demos should run in the browser.
- To make these interactive demos effective:

  1. Show, don’t tell.
  2. Be intriguing at first glance.
  3. Don’t show too much too soon.
  4. Build in support for teachers.