CS354R

DR SARAH ABRAHAM

MACHINE LEARNING IN GAMES

WHAT IS ML?

- Machine learning is a broad term to describe methods that allow computer system to learn/improve without fully specifying how to accomplish this
    - Contrasted with classical AI methods that use fully specified knowledge/rules to determine how to accomplish a task
- Wide range of techniques including simple statistical models (e.g. PCA analysis) to complex optimization functions (deep networks)

OPTIMIZATION FOR CLASSIFICATION

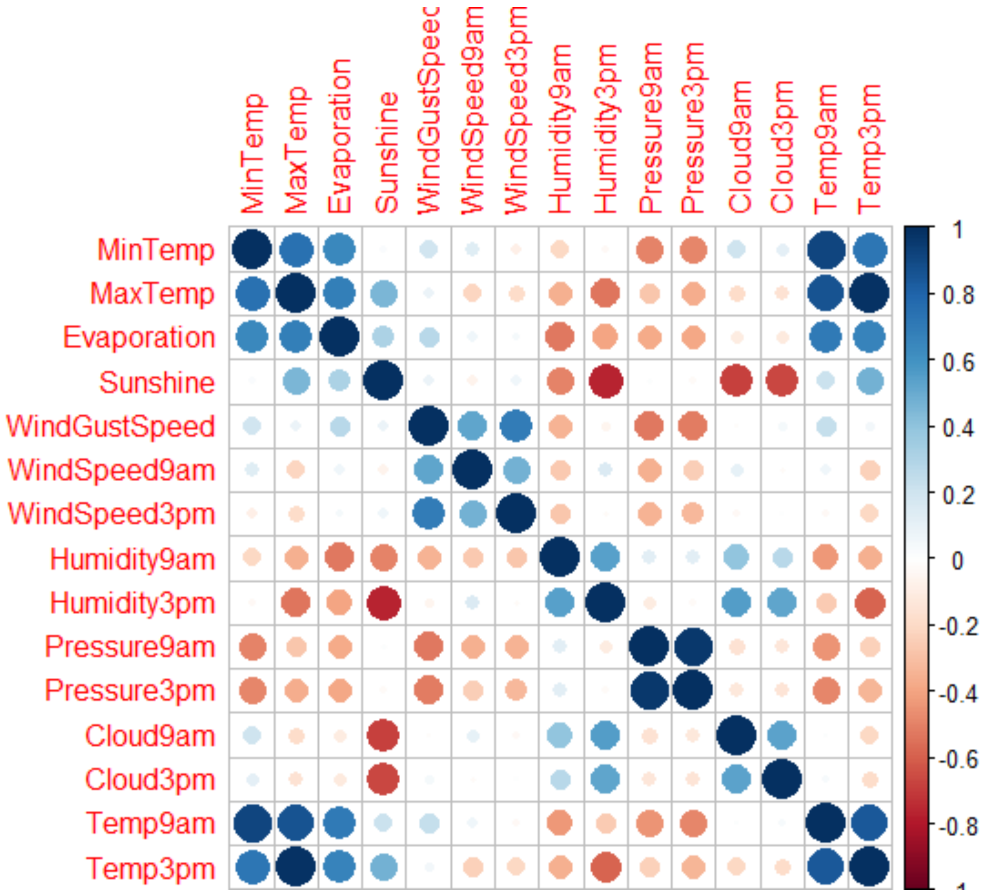
- Estimate the likelihood of an object belonging to a given category

handwritten number recognition

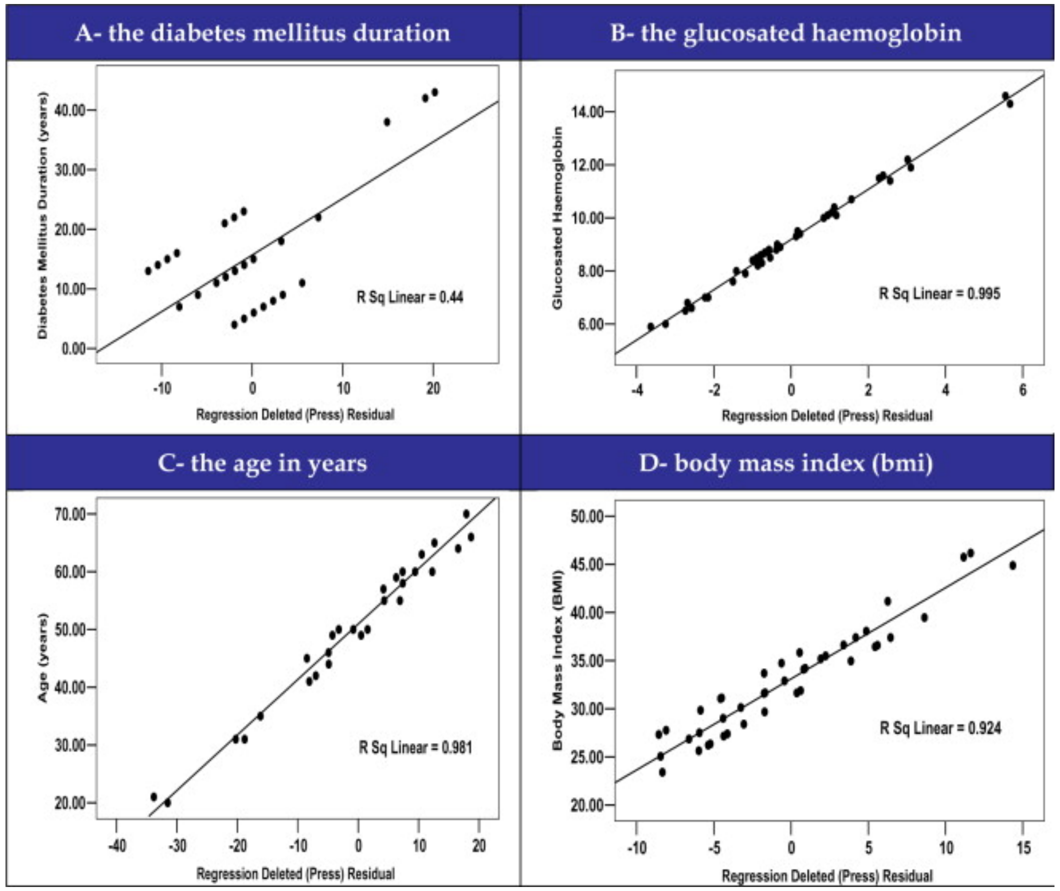
image segmentation

OPTIMIZATION FOR REGRESSION

- Minimize error (difference) between a predicted value and a target value



Weather forecasting



Health risks

EXAMPLE: GRADIENT DESCENT

- Common iterative technique for finding an optimum value
- Basic idea:
  - Error can be quantified as a loss function (e.g. mean square error, mean absolute error etc)
  - This loss is differentiable (i.e. 1st derivative tells us how much the function has changed, 2nd derivative tells us speed of this change)
  - Possible to explore a parameterized function by following this gradient to some minimum

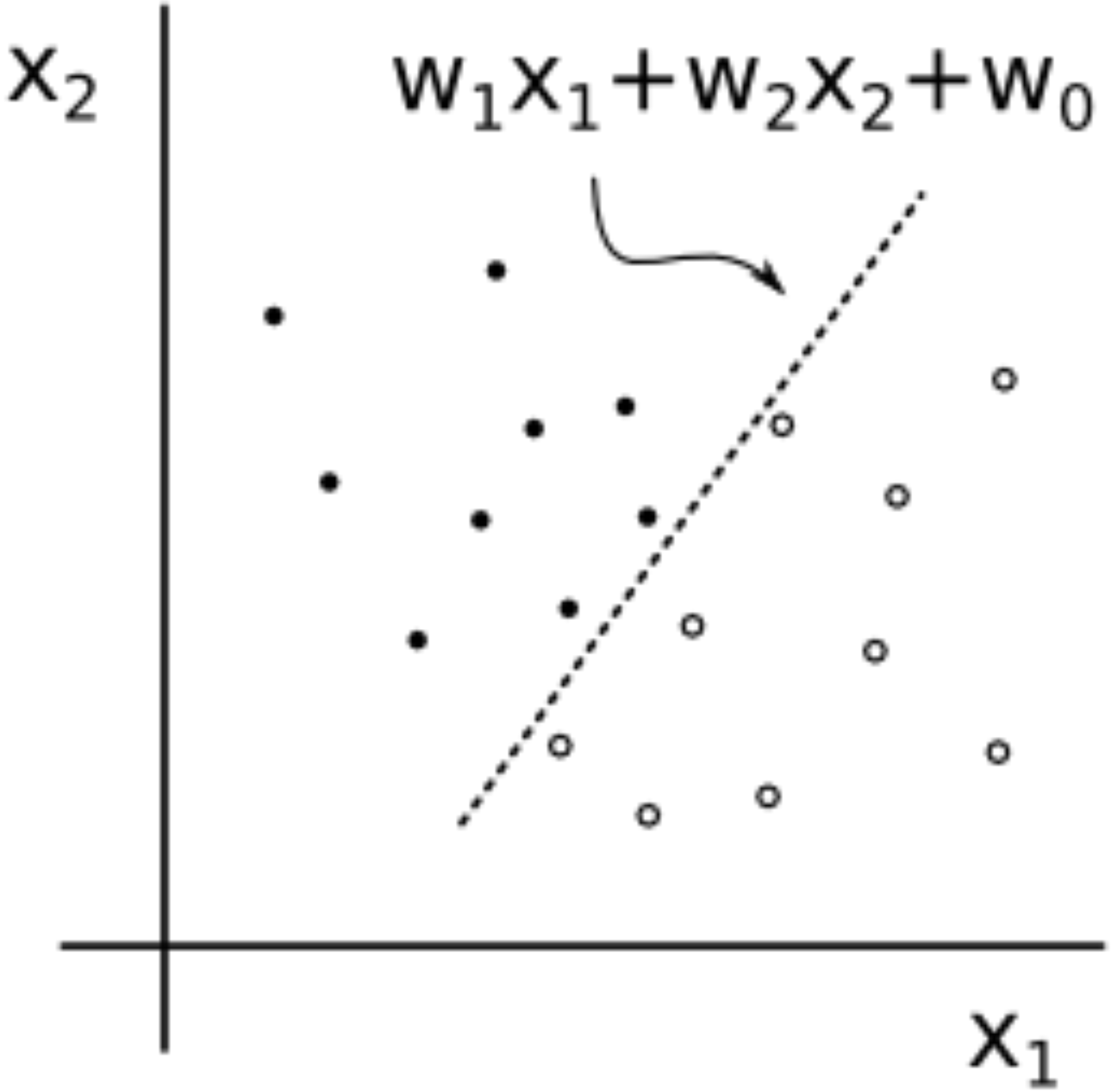
LIMITATIONS

- Local minimums
- Overfitting
- Linear in nature

NEURAL NETWORKS

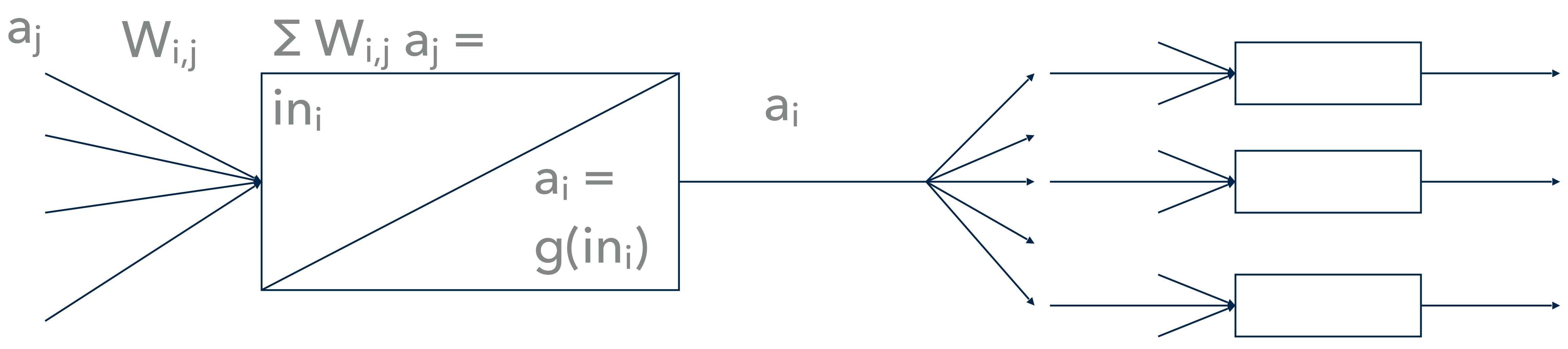
- Inspired by natural decision making structures (real nervous systems and brains)
- If you connect lots of simple decision making pieces together, they can make more complex decisions
  - Compose simple functions to produce complex functions
- Take multiple numeric input variables
- Produce multiple numeric output values
- Threshold outputs to turn them into discrete values
- Map discrete values onto classes, and you have a classifier!
  - Also work as approximation functions

DECISION BOUNDARY



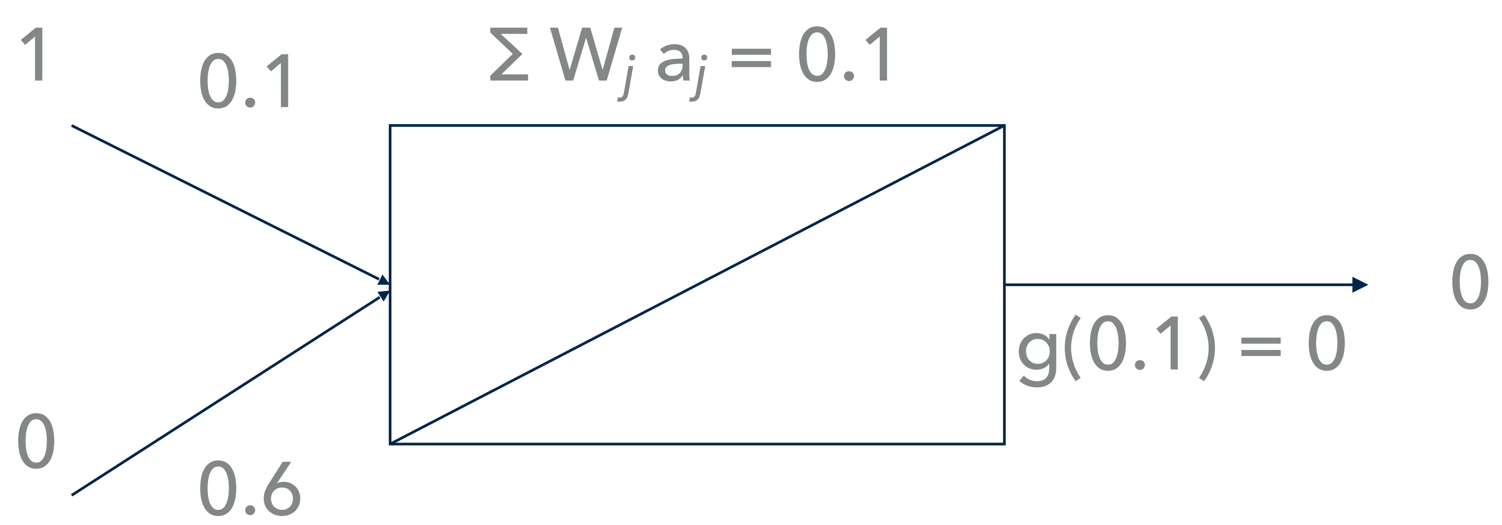
SIMULATED NEURON: PERCEPTRON

- Inputs (a_j) from other perceptrons with weights ($W_{i,j}$)
  - Learning occurs by adjusting the weights
- Perceptron calculates weighted sum of inputs (in_i)
- Threshold function calculates output (a_i)
  - Step function (if $in_i > t$ then $a_i = 1$ else $a_i = 0$)
  - Sigmoid $g(a) = 1/(1 + e^{-x})$
- Output becomes input for next layer of perceptron



PERCEPTRON EXAMPLE

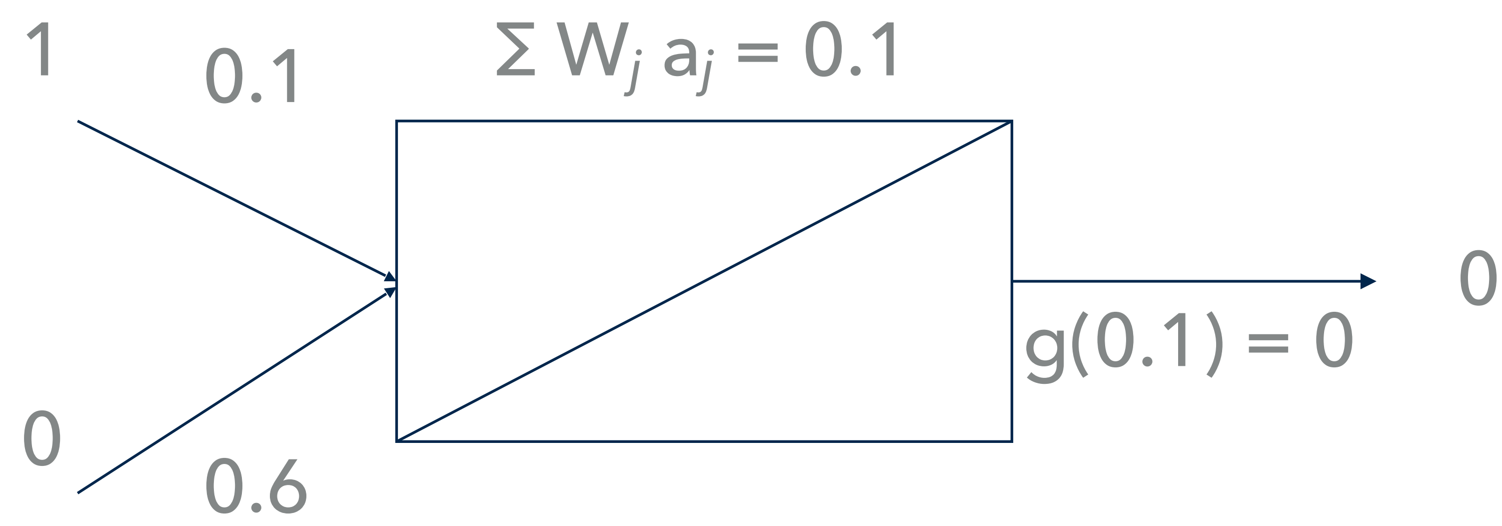
- Single perceptron to represent OR
    - Two inputs
    - One output (1 if either inputs is 1)
    - Step function (if weighted sum > 0.5 output a 1)
- Initial state (below) gives error on (1,0) input
    - Need more training…



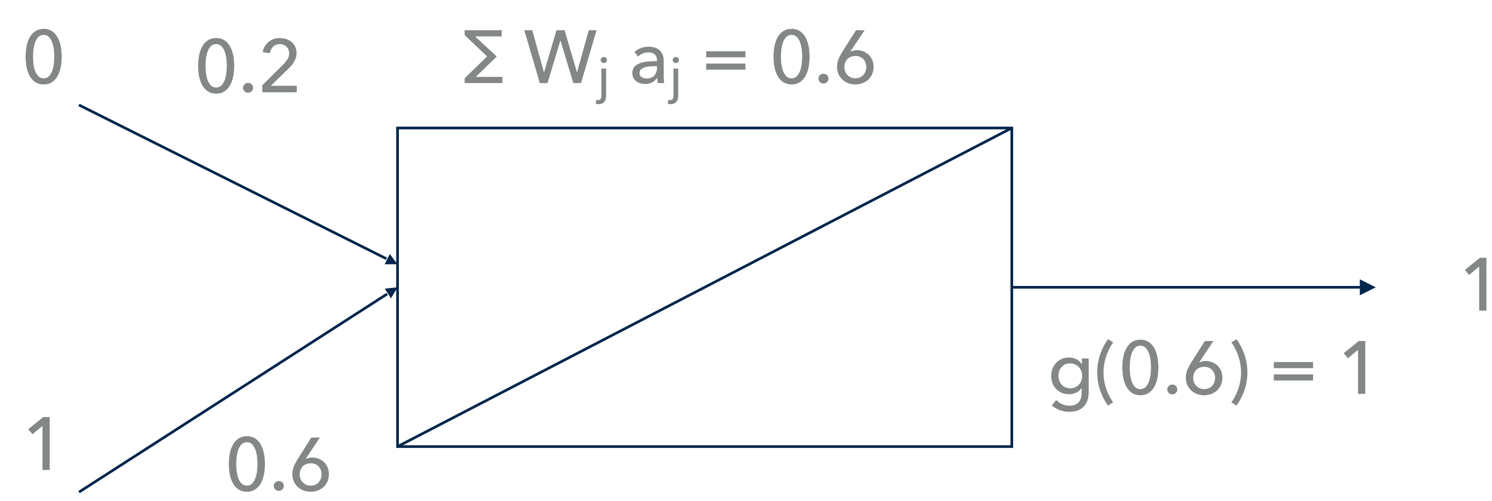
LEARNING NEURAL NETWORKS

- Learning from examples
  - Examples consist of correct output t, and predicated output o
- Learn if network's output doesn't match correct output
  - Adjust weights to reduce difference
  - Learning rate only change weights a small amount (η)
- Basic perceptron learning
  - $W_{i,j} = W_{i,j} + \eta(t\text{-}o)a_j$
  - If output is too high (t-o) is negative so $W_{i,j}$ will be reduced
  - If output is too low (t-o) is positive so $W_{i,j}$ will be increased
  - If a_j is negative the opposite happens

NEURAL NET TRAINING

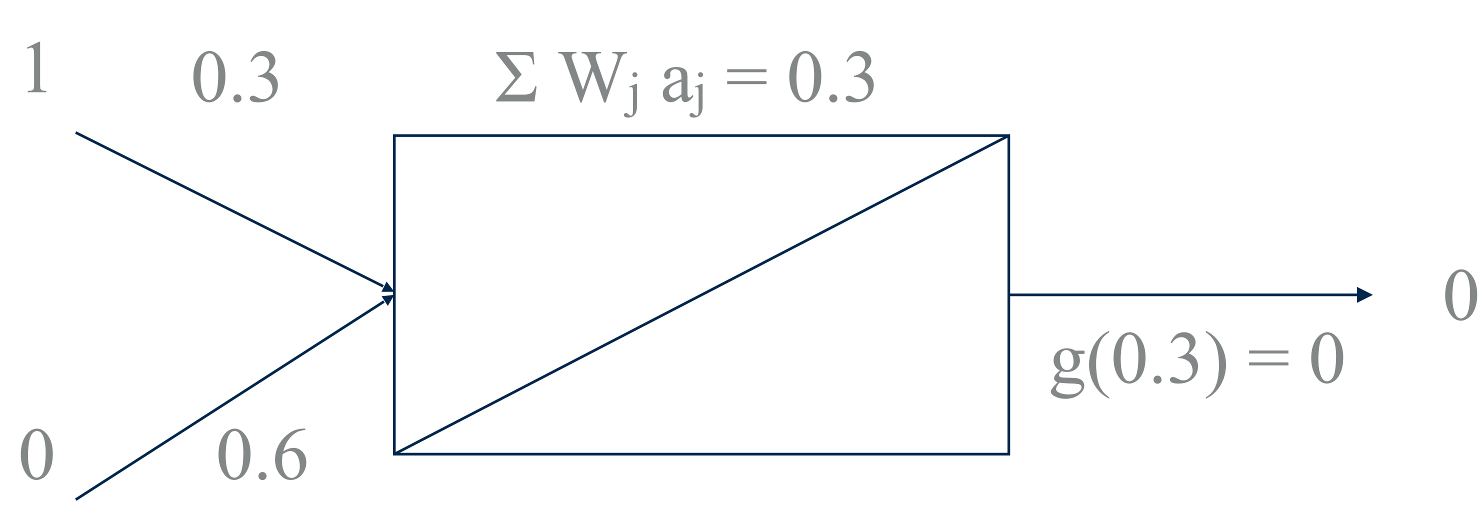


- $W_j = W_j + \eta(t\text{-}o)a_j$
- W1 = 0.1 + 0.1(1-0)1 = 0.2
- W2 = 0.6 + 0.1(1-0)0 = 0.6
- After this step, try (0,1)→1 example
  - No error!



NEURAL NET TRAINING

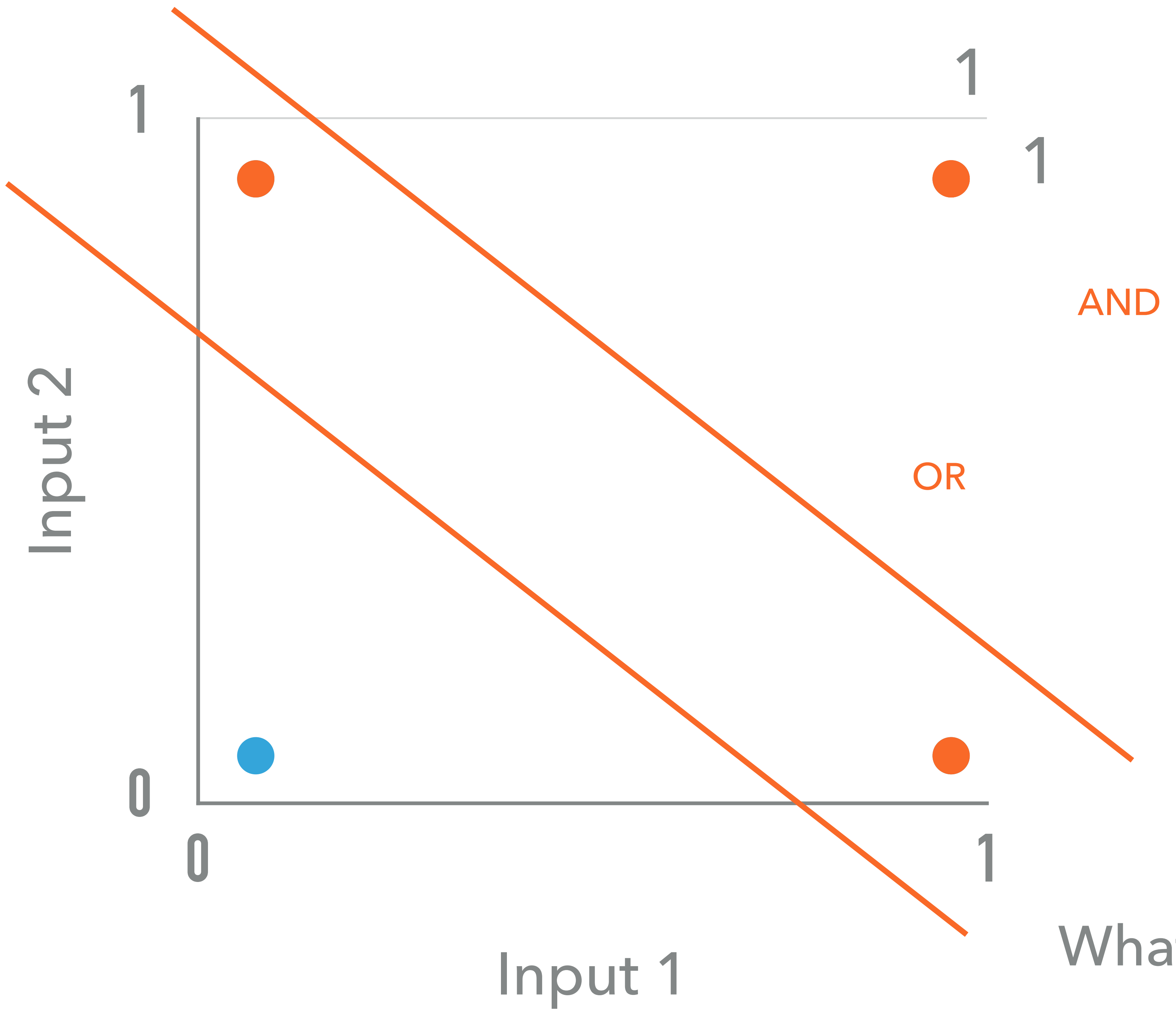
- Try (1,0)→1 example
  - Still an error, so training occurs
- W1 = 0.2 + 0.1(1-0)1 = 0.3
- W2 = 0.6 + 0.1(1-0)0 = 0.6
- And so on…



PERCEPTRON INSIGHTS

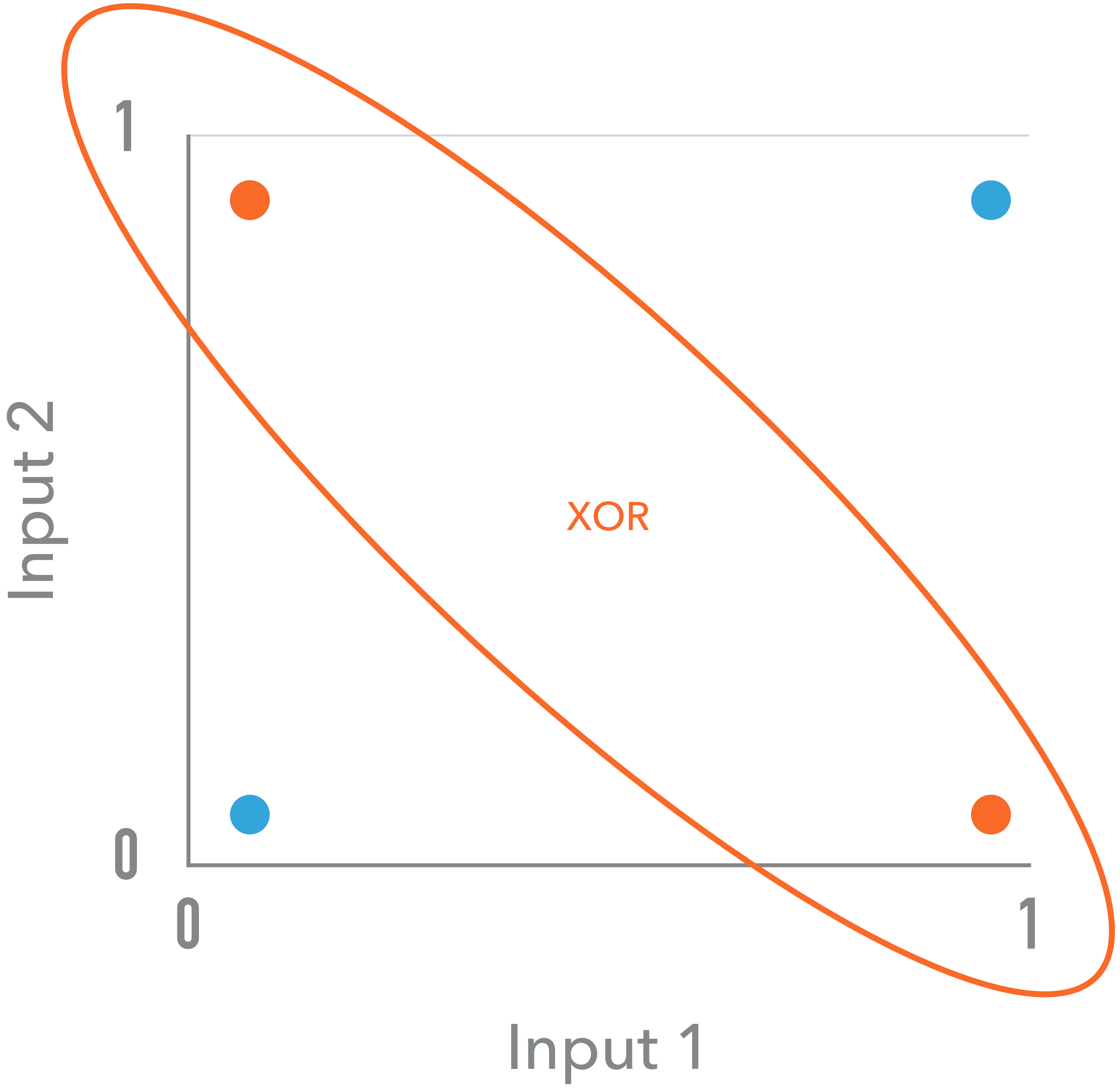
- Perceptron evaluates linear combination of weighted inputs with a bias to create a decision boundary
- A neat trick but ultimately very limited…

PERCEPTRON PROBLEM...



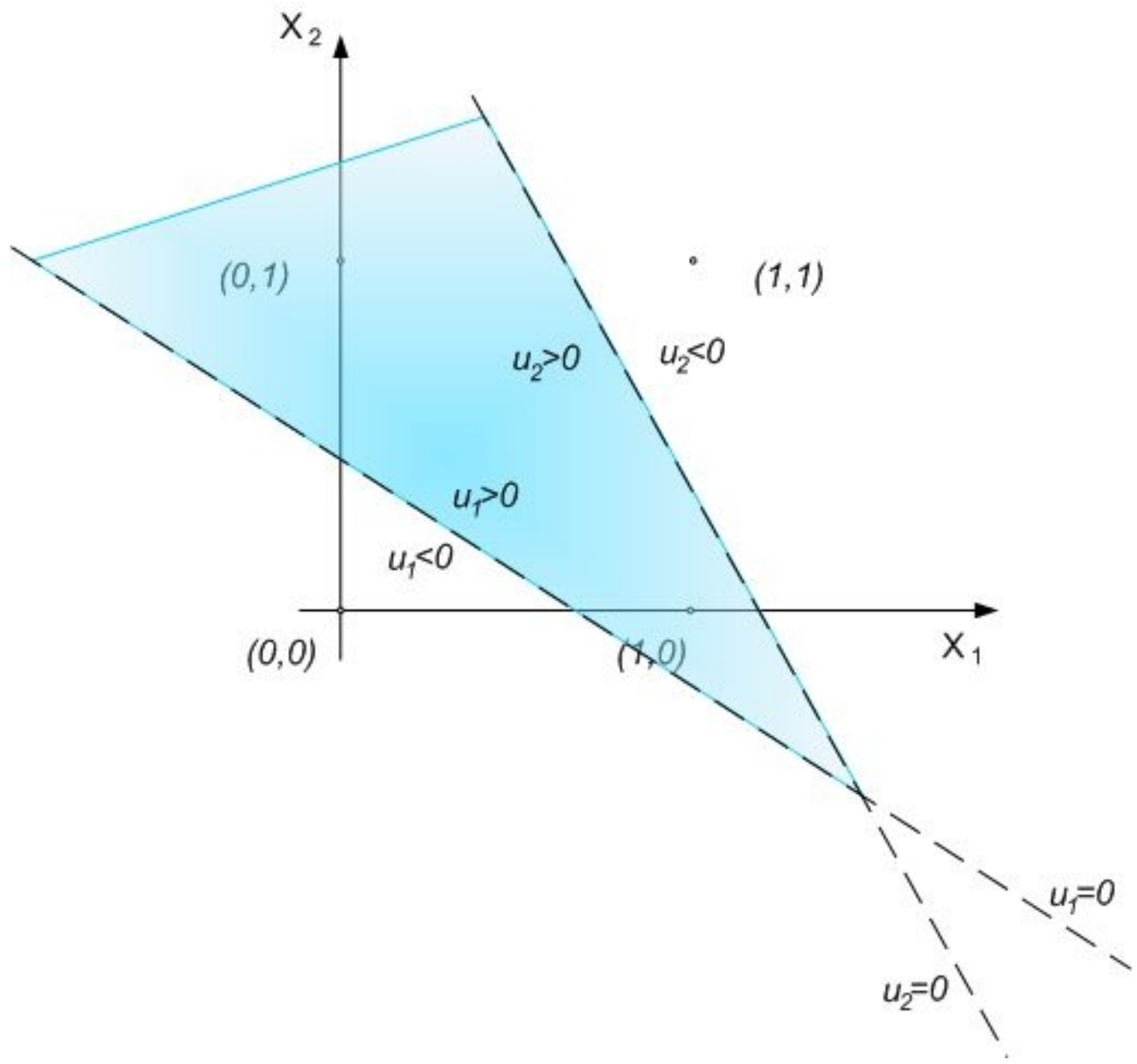
What about XOR?

REPRESENTING XOR



SOLVING XOR

- Use a combination of perceptrons!



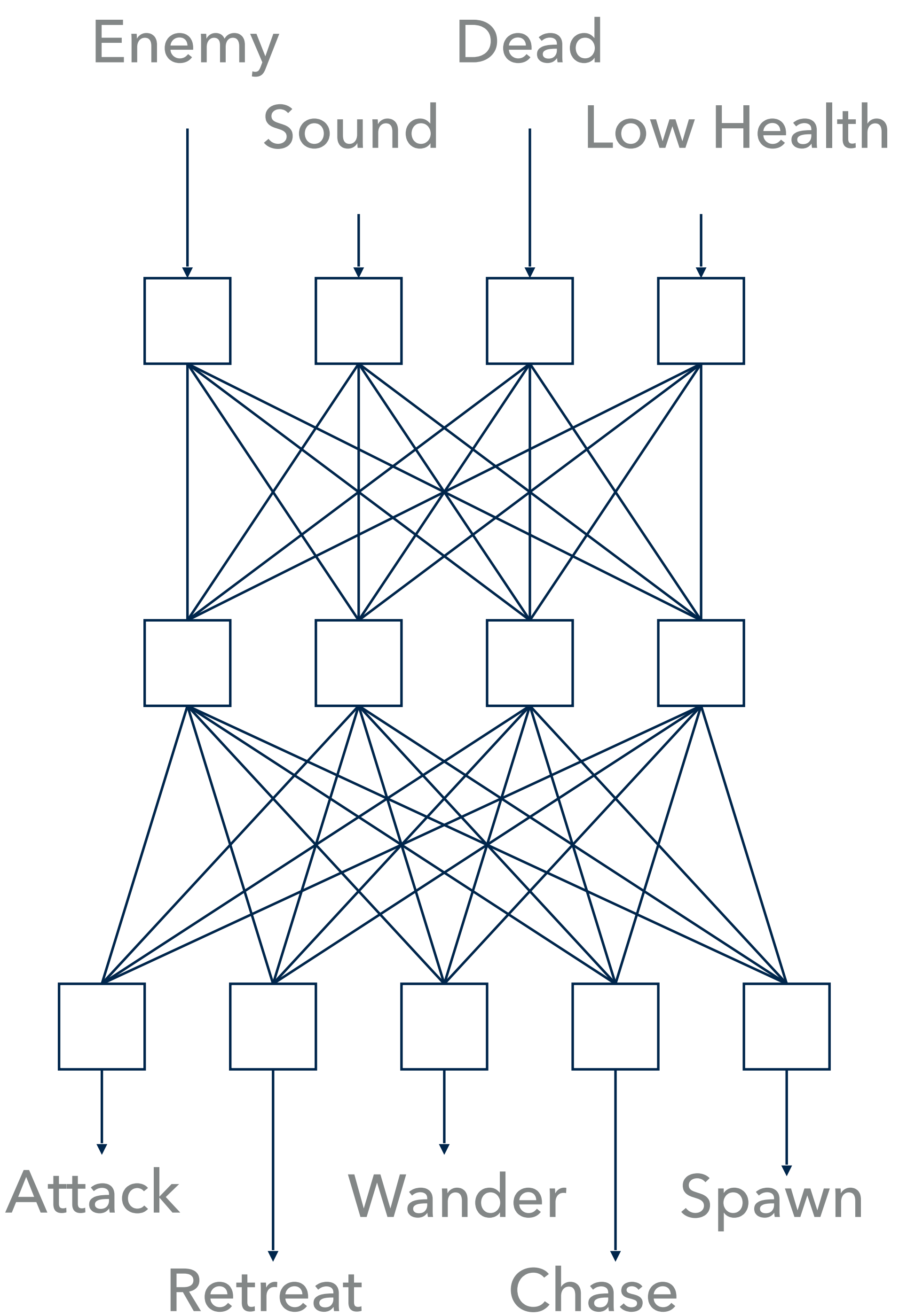
(AGH University Virtual Laboratory of AI)

NEURAL NETWORK STRUCTURE

- Perceptron are usually organized in layers
  - Input layer: takes external input
  - Hidden layer(s)
  - Output layer: external output
- Feed-forward vs. recurrent
  - Feed-forward: outputs only connect to later layer
  - Learning is easier
  - Recurrent: outputs can connect to earlier layers or same layer
    - Internal state

NEURAL NETWORK FOR QUAKE

- Four input perceptron
  - One input for each condition
- Four perceptron hidden layer
  - Fully connected
- Five output perceptron
  - One output for each action
  - Choose action with highest output
  - Or, probabilistic action selection
    - Choose at random weighted by output



NEURAL NETWORKS AS AI

- Forza
- Supreme Commander 2
- Black & White

ISSUES WITH ML-BASED AI

- Hard to control
  - Difficult for designers to tune
  - Difficult to debug
- Most ML-based AI in games is more for researching/testing ML concepts than creating a player experience
  - Creating game agents using reinforcement learning, genetic algorithms, neural networks, deep reinforcement learning, etc, etc, etc…

EXAMPLE: GRAN TURISMO

- Outracing champion Gran Turismo drivers with deep reinforcement learning
  - https://www.nature.com/articles/s41586-021-04357-7
- Created using Deep Reinforcement Learning techniques
  - https://youtu.be/V2Is7A5BNuw?si=28K5JvtJYpPUprQ_&t=119
  - https://www.youtube.com/watch?v=AFeoN_kOuJg

DEEP LEARNING AND REINFORCEMENT LEARNING

- Deep learning refers to a category of neural network techniques with a large (deep) number of layers through which data is transformed
    - Popular technique for *unsupervised* learning problems that exist in a high dimensional space with access to lots of data
- Reinforcement learning allows an agent to explore a complex environment and learn actions and behaviors based on a reward function
    - Popular technique in robotics where the state is too large to allow for supervised learning

DEEP LEARNING IN ANIMATIONS

- Rich area of research in graphics!
- Animation synthesis uses large amounts of example data combined with deep learning techniques to optimize novel behaviors
- Uses in:
  - Fluid flow approximation
  - Character animation synthesis

EXAMPLE: EA SPORTS MMA

- Neural Animation Layering for Synthesizing Martial Arts Movements
  - https://80.lv/articles/ea-studies-the-use-of-deep-learning-for-combat-animations/
- Allows mixing and blending of animations to synthesize novel animations
  - https://www.youtube.com/watch?v=SkJNxLYNwN0

EXAMPLE: FINAL FANTASY VII REBIRTH

- Lip-sync ML for generating lip motions based on language in real time
  - http://www.jp.square-enix.com/tech/library/pdf/LipSyncML_SIGGRAPH_Talks_2024_slides.pdf
- [Demo]

DLSS REVISITED

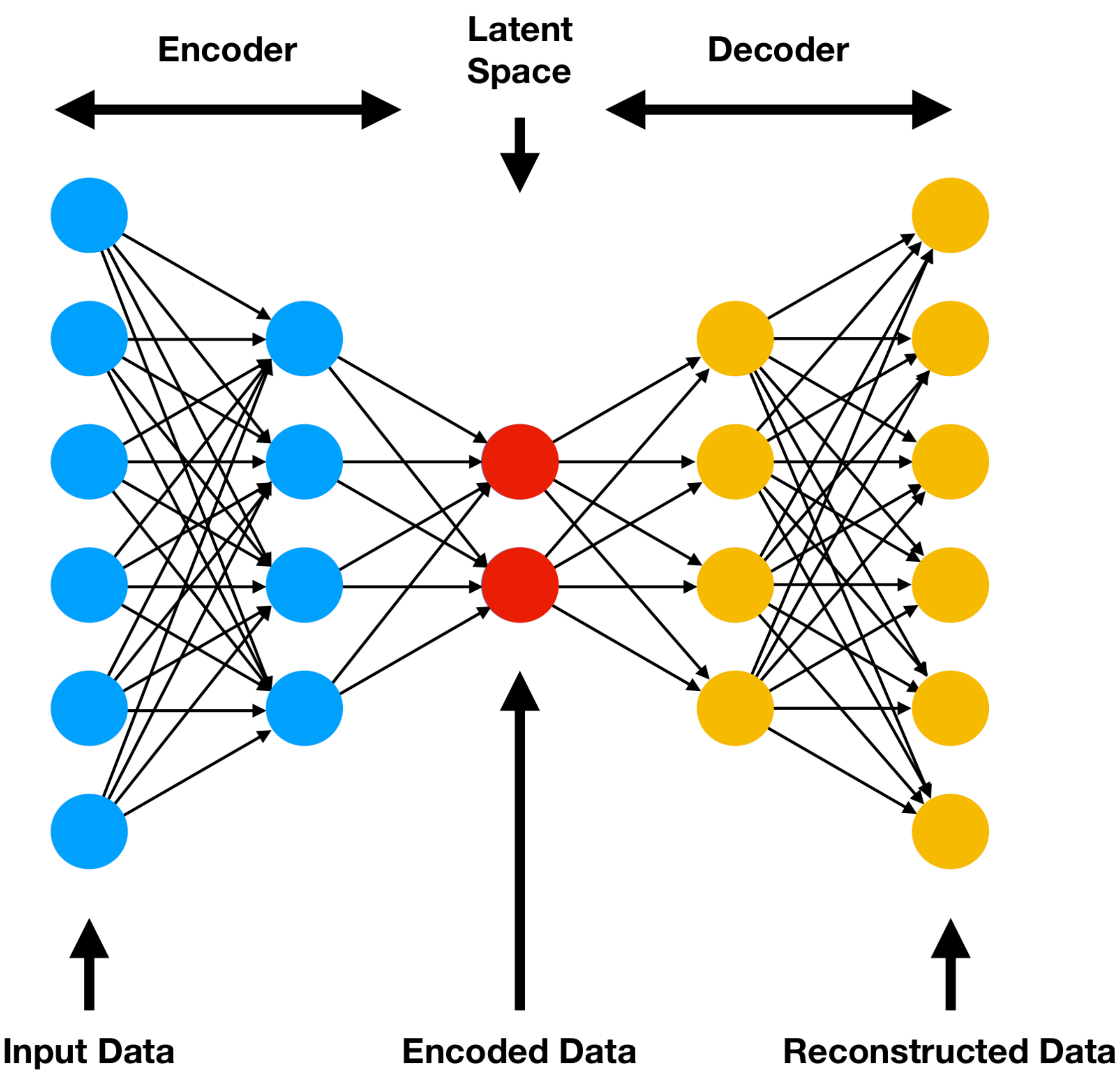
- Deep Learning Super Sampling
  - Upscaling techniques that allow graphics pipeline to run at lower resolutions before upsampling the final outputted image
  - DLSS 3 allows for full frame generation
- DLSS is proprietary NVidia technology
  - FSR is open source AMD version
- Assumed for all modern Triple A (and many Double A) titles
  - https://www.youtube.com/watch?v=GkUAGMYg5Lw&t=386s

SIDE NOTE: GENERATIVE TECHNIQUES

- Use of a **latent feature space** that allows sets of items to be embedded such that proximity in this space reflects similarity
  - Smaller than the initial feature space
  - Possible to **encode** objects into a latent space of information then **decode** that object to recreate its representation

AUTOENCODERS

- Style of neural network architecture that encode data to this latent space then decode data to recreate this data
- Basis of many applications in NLP, image generation, computer vision etc



SIDE NOTE: NERFS

- Neural radiance fields
  - Allow reconstruction of a 3D scene from a 2D image using deep learning
- Fundamentally a rendering technique
  - Returns a color value at some (x, y, z) coordinate based on the calculated incoming radiance
- Not super applicable to games as they contain no collision data and honestly I think they're overhyped so that's all I'll say :)